(19)日本国特許庁（ＪＰ）　(12)**公開特許公報**（Ａ）　(11)特許出願公開番号

特開平10－178450

(43)公開日　平成10年(1998)６月30日

(51)Int.Cl.[6]	識別記号	ＦＩ		
Ｈ０４Ｌ　12/56		Ｈ０４Ｌ	11/20	１０２Ｚ
Ｇ０９Ｃ　1/00	６６０	Ｇ０９Ｃ	1/00	６６０Ｅ
Ｈ０４Ｌ　12/46		Ｈ０４Ｌ	11/00	３１０Ｃ
12/28			11/26	
12/22			9/00	６８５

審査請求　未請求　請求項の数15　ＯＬ　（全 29 頁）　最終頁に続く

(21)出願番号　特願平9－290739

(22)出願日　平成９年(1997)10月23日

(31)優先権主張番号　０８／７３８１５５

(32)優先日　1996年10月25日

(33)優先権主張国　米国（ＵＳ）

(71)出願人　590002873
ディジタル　イクイプメント　コーポレイション
アメリカ合衆国　マサチューセッツ州　01754－1418　メイナード　パウダー　ミル　ロード　111

(72)発明者　ケニス　エフ　オルデン
アメリカ合衆国　マサチューセッツ州　01505ボイルストン　クロス　ストリート　188

(74)代理人　弁理士　中村　稔　（外６名）

最終頁に続く

(54)【発明の名称】　フレームを捕獲、カプセル化及び暗号化するための擬似ネットワークアダプタ

(57)

２５９　Ｄ

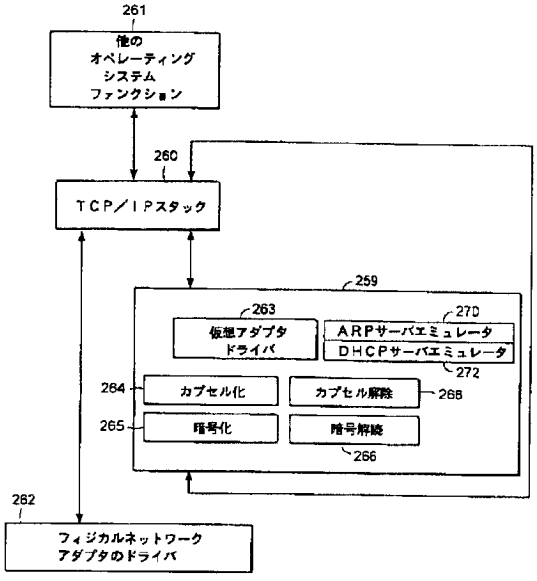
２７２　ＡＲＰ

ＧＷ

２６０

２６５

２６８



[redacted]

1

[redacted]

4

[redacted]

1

[redacted] 2 [redacted]

7

[redacted]

3　　4

[redacted]と[redacted]ア
[redacted]ア[redacted]
[redacted]
[redacted]
[redacted]に[redacted]上
[redacted]上記[redacted]
[redacted]パ[redacted]
[redacted]信[redacted]
[redacted]生[redacted]ハ
[redacted]ッ[redacted]　9
[redacted]方法[redacted]　2
[redacted]
[redacted]た[redacted]
[redacted]る[redacted]
[redacted]　1
[redacted]に記[redacted]　2
[redacted]
[redacted]の[redacted]上
[redacted]記[redacted]
[redacted]と[redacted]の
[redacted]受[redacted]取
[redacted]つ[redacted]
[redacted]ア[redacted]上
[redacted]ア記[redacted]
[redacted]パ[redacted]
[redacted]
[redacted]ハ[redacted]
[redacted]ア[redacted]
[redacted]
[redacted]取
[redacted]つ[redacted]
[redacted]た[redacted]
[redacted]カ[redacted]
[redacted]と備[redacted]　9
[redacted]
[redacted]
[redacted]イ[redacted]ア
[redacted]
[redacted]振[redacted]　9
[redacted]の[redacted]
9[redacted]　40
[redacted]ラ[redacted]　0001
[redacted]
[redacted]程[redacted]ア[redacted]
[redacted]ッ[redacted]
[redacted]号[redacted]
[redacted]　0002
[redacted]
[redacted]管[redacted]
[redacted]
[redacted]か[redacted]

[redacted]

1

[redacted]

A

[redacted]

0009

[redacted]

B

[redacted]

B

[redacted]

IP

[redacted]

lied Cryptography

[redacted]

IP

[redacted]

[redacted]

Protoco[redacted]lgorithms [redacted]

1994 [redacted]

[redacted]

0012 [redacted]

0018 [redacted]TCP 22

[redacted] 26 OSI

[redacted]S の [redacted] IP 26

[redacted] 1[redacted] OSI

[redacted] 10

[redacted] AR

[redacted] IP

[redacted] MAC

[redacted] I[redacted] ARP IP

[redacted] P P IP

[redacted]puter Networks) [redacted] ARP 2

88 [redacted] 要[redacted] ARP

[redacted] IP

[redacted] 5 [redacted] 28 ARP

[redacted] IP

[redacted] ARP ARP

[redacted] 発[redacted] 30 ARP

[redacted] IP

[redacted] IP

[redacted] IP

[redacted] ARP

[redacted] IP

[redacted] TCP 接[redacted] 要[redacted] ARP

[redacted] IP

[redacted] 対[redacted] ARP

[redacted] 1[redacted] TCP IP

(Interwor[redacted] IP

[redacted] DHCP

[redacted] IP DHC

[redacted]

52

I要[redacted]　　2　　5

サ[redacted]　　N2 60

[redacted]　　3

[redacted]要[redacted]　　D 62

[redacted]IP[redacted]

[redacted]　　60

[redacted]

[redacted]

[redacted]　　N2 60

[redacted]

[redacted]　　N1 48

[redacted]A[redacted]　　N2 60

[redacted]　　3

[redacted]　　48　　60

[redacted]

[redacted]　　A 4

[redacted]　　D 62

[redacted]　　A 46　　A

[redacted]　　4　　B

[redacted]　　5　　C　　D 6

[redacted]

[redacted]

[redacted]　　A　　B　　B

[redacted]　　C　　D

[redacted]

[redacted]

[redacted]　　PC

[redacted]

[redacted]　　5　　3

[redacted]　　A 16　　A　　D

[redacted]D

[redacted]　　4

[redacted]段[redacted]　　A　　D

[redacted]

[redacted]3

[redacted]　　A

[redacted]

[redacted]

[redacted]8　　[redacted]e arc

[redacted]　　TCP IP

[redacted]　　DNS

[redacted]る　　C

[redacted]B[redacted]4　　A B C

[redacted]

[redacted]

[redacted]

TCP [redacted] B C

C D

[redacted] B

[redacted] A [redacted] 80

[redacted] D

[redacted]

[redacted] REQUEST

[redacted] RESPONSE

[redacted]

[redacted]

[redacted]

[redacted]

[redacted] 90

[redacted] A

[redacted] 92

[redacted] A 5

[redacted] 5

[redacted] 1

[redacted] D

[redacted] PC

[redacted]は[redacted]ドは[redacted]　122
[redacted]
[redacted]と指[redacted]8　9　[redacted]UEST
[redacted]は[redacted]
[redacted]ドは[redacted]6
[redacted]
[redacted]
[redacted]は[redacted]
[redacted]の[redacted]
[redacted]る[redacted]
ドは[redacted]
[redacted]102[redacted]
[redacted]Yは[redacted]ドは[redacted]3　D
[redacted]
[redacted]
[redacted]ONS[redacted]ドは[redacted]130
[redacted]例[redacted]ド内[redacted]138
[redacted]
[redacted]換[redacted]
[redacted]へ[redacted]
[redacted]
[redacted]1換[redacted]ド　132
[redacted]ドは[redacted]は[redacted]
[redacted]の[redacted]
[redacted]の[redacted]
[redacted]等[redacted]ドは[redacted]134
[redacted]ドは[redacted]
[redacted]む[redacted]ドは[redacted]136
[redacted]
[redacted]
[redacted]
[redacted]
[redacted]は[redacted]138
[redacted]
[redacted]3[redacted]
[redacted]端[redacted]7
[redacted]5
[redacted]段[redacted]
[redacted]0[redacted]8　RESPONSE
[redacted]
[redacted]ドは[redacted]150
[redacted]内[redacted]ドは[redacted]152
[redacted]
[redacted]ONSE　1
[redacted]は[redacted]
[redacted]ドは[redacted]156
[redacted]ド内[redacted]163
3　[redacted]る[redacted]
[redacted]
[redacted]

[redacted]
[redacted]　3
[redacted]
[redacted]　A
[redacted]　16
[redacted]
[redacted]
[redacted]　190
[redacted]　5
[redacted]
[redacted]
[redacted]
[redacted]
[redacted]　187
[redacted]　190
[redacted]
[redacted]　187
[redacted]　188
[redacted]　IP
[redacted]
[redacted]
[redacted]
[redacted]　190
[redacted]
[redacted]
[redacted]
[redacted]
[redacted]　191
[redacted]　187
[redacted]　192
[redacted]
[redacted]
[redacted]
[redacted]　10
[redacted]
[redacted]
[redacted]

[redacted]1 [redacted] は [redacted] 211

[redacted] [redacted] に [redacted] 212

[redacted] 12 [redacted] は [redacted] 213

[redacted] [redacted] 接[redacted] 1 TCP 21

[redacted] は [redacted] T

[redacted] D [redacted] に [redacted] CP 195

[redacted] [redacted] の [redacted] 203

[redacted] 2[redacted] は [redacted] 204

[redacted] [redacted]

[redacted] [redacted] [redacted] 205

[redacted] 者[redacted] に [redacted] 206 206

[redacted] の [redacted] 2

[redacted] 遷移 TC[redacted]

[redacted] は [redacted] に [redacted] 208

[redacted] [redacted] は [redacted] 12 3

[redacted] の [redacted] バ内[redacted] 14 253

[redacted] と [redacted]

[redacted] 接[redacted] [redacted] 217

[redacted] [redacted] 接[redacted] と [redacted] TCP 218

[redacted]

[redacted] は [redacted]P の [redacted] 接[redacted] は [redacted] TCP 218

[redacted] 答[redacted]

[redacted] に [redacted] [redacted] 接[redacted] C TCP

[redacted] 107 [redacted]

[redacted] [redacted] [redacted] 後

[redacted] の [redacted] 接[redacted] [redacted] TCP

[redacted] に [redacted] に [redacted] 219 219

[redacted] は [redacted] は [redacted] 220

[redacted]

[redacted] は [redacted] に [redacted]08 は [redacted] 219

[redacted] 接[redacted] 信[redacted] ラ が [redacted] 234

[redacted] 合[redacted] に [redacted] 230

[redacted] る [redacted] の [redacted] 230

[redacted] [redacted]

[redacted] [redacted] 接[redacted] は [redacted] 231

[redacted] 接[redacted] [redacted] 232

[redacted] 194 に [redacted]

[redacted] は [redacted] [redacted] は [redacted] 221

[redacted] は [redacted] 2[redacted]

[redacted] [redacted] は [redacted] 222

[redacted] [redacted]

[redacted] [redacted] は [redacted] 223

[redacted] [redacted]

[redacted] [redacted]

[redacted] 3 [redacted]

[redacted] は [redacted]

[redacted] 2[redacted] は [redacted]

[redacted] [redacted]

[redacted]　27[redacted]　[redacted]241
[redacted]の[redacted]は[redacted]　243
[redacted]
[redacted]る[redacted]
[redacted]63　[redacted]に[redacted]に244　244
[redacted]241
[redacted]2切[redacted]に　246
[redacted]
[redacted]　[redacted]は[redacted]14　24
[redacted]に[redacted]　24
[redacted]33バ　[redacted]は[redacted]　251
[redacted]23[redacted]とバとの[redacted]　253
[redacted]は[redacted]バ[redacted]バ[redacted]　253
[redacted]は[redacted]247　CPU
[redacted]は[redacted]　I O
[redacted]バ[redacted]バ　253
[redacted]5は[redacted]と[redacted]　254
[redacted]ッ[redacted]8[redacted]1　257
[redacted]ズ[redacted]イ[redacted]2　[redacted]　LAN256
[redacted]9は[redacted]バは[redacted]　253
230　[redacted]イ[redacted]と[redacted]　LAN256　2
[redacted]13　か[redacted]　[redacted]　252
[redacted]は[redacted]　252
[redacted]2[redacted]は[redacted]51
[redacted]は[redacted]サ[redacted]8[redacted]　25
[redacted]と[redacted]　2
[redacted]は[redacted]は[redacted]　251
[redacted]で[redacted]アカ[redacted]と[redacted]　250
[redacted]
[redacted]　[redacted]は[redacted]14
[redacted]の[redacted]　[redacted]内[redacted]249
[redacted]238　イ[redacted]は[redacted]
[redacted]　プ[redacted]と[redacted]
[redacted]36[redacted]
[redacted]　[redacted]イ[redacted]と　248
[redacted]　[redacted]と[redacted]　247
[redacted]サ[redacted]バとの[redacted]
[redacted]　[redacted]と[redacted]バとの[redacted]　253
[redacted]　[redacted]40
[redacted]　239　[redacted]　[redacted]は[redacted]5　14
[redacted]は[redacted]プ[redacted]の[redacted]
[redacted]に[redacted]　例[redacted]1　15
[redacted]240[redacted]に[redacted]14　24
[redacted]　は[redacted]
[redacted]　図[redacted]　259
[redacted]　は[redacted]と[redacted]　263
[redacted]2[redacted]と[redacted]　264　265
[redacted]に[redacted]と[redacted]と[redacted]　268
[redacted]　[redacted]と[redacted]

[redacted]59 [redacted]R[redacted]は[redacted]286 T
[redacted]から[redacted]フ 282
[redacted]も[redacted] [redacted]288
[redacted]
[redacted]0は[redacted]、[redacted]と[redacted]2
[redacted]と[redacted] は[redacted]から[redacted]296
[redacted]288 TCP IP
[redacted] T[redacted] [redacted]2 282
[redacted]の[redacted] [redacted]D[redacted]16 280
[redacted]に[redacted]は[redacted]296
[redacted]288
[redacted]は[redacted]298 300
[redacted]の[redacted]と[redacted]は[redacted]300
[redacted]の[redacted]CP IP 282 TCP IP
[redacted]に[redacted]314
[redacted]5[redacted]1
[redacted]に[redacted]2[redacted]5[redacted]
[redacted]2[redacted]9[redacted]
[redacted]の[redacted]TCP[redacted]280
[redacted]に[redacted]ARP 304 DHC
[redacted]306 ARP
[redacted]は[redacted]DHC[redacted]302 ARP
[redacted]304 DHCP 30
[redacted]から[redacted]296
[redacted]ら[redacted]282
[redacted]は[redacted]
[redacted]NDI
[redacted]S[redacted]TPC IP
[redacted]2[redacted]2
[redacted]IP[redacted]に[redacted]16
[redacted]は[redacted]TCP IP
[redacted]に[redacted]TCP IP
[redacted]は[redacted]
[redacted]16
[redacted]16[redacted]は[redacted]28[redacted]10 TCP IP
[redacted]
[redacted]は[redacted]90[redacted]TCP IP
[redacted]に[redacted]2
[redacted]は[redacted]94[redacted]16
[redacted]TCP IP[redacted]282
[redacted]の[redacted]
[redacted]TCP IP
[redacted]は[redacted]82
[redacted]6[redacted]
[redacted]8[redacted]
[redacted]TC
[redacted]P IP[redacted]は[redacted]282
[redacted]さ[redacted]は[redacted]30[redacted]286

[redacted]280 [redacted]
[redacted]と[redacted] [redacted]286は[redacted]ア 326
[redacted] [redacted]サ[redacted] 応答[redacted]P ARP
[redacted]パ[redacted]要[redacted] ARP
[redacted]送[redacted]TC[redacted] IP
[redacted]IP[redacted]
[redacted]は[redacted]応答[redacted] IP
[redacted]所[redacted]
[redacted]ア[redacted]は[redacted] AR
[redacted]応答[redacted]
[redacted]
[redacted]かP IP[redacted]イは[redacted]イ NDIS
[redacted]ア[redacted] TCP IP
[redacted]ア[redacted] TCP IP
[redacted]応答[redacted]
[redacted]
[redacted]大[redacted]は[redacted] 330
[redacted]は[redacted] IP
[redacted]要[redacted]DHCP
[redacted]Tは[redacted]の[redacted] 330
[redacted]2から[redacted]ア[redacted]が[redacted]大[redacted]3の[redacted]後 330
[redacted]にTCPに[redacted]大[redacted]が[redacted]282 334
[redacted]か[redacted]大[redacted]は[redacted]ア 334
[redacted]サ[redacted] DHCP DHCP
[redacted]応答[redacted]の[redacted]DHCP
[redacted]は[redacted]RCP REL 328
[redacted]ラにお[redacted]応答[redacted] DHCP
[redacted]
[redacted]17れ[redacted] IP
[redacted]
[redacted]
[redacted] TC
[redacted]ア[redacted]IP/ TCP IP
[redacted]ア[redacted]応答[redacted] DHCP
[redacted]大[redacted]は[redacted]か[redacted]
[redacted]ア[redacted]
[redacted]大[redacted]は[redacted]ア 334
[redacted]ア[redacted]
[redacted]スは[redacted]号[redacted]
[redacted]れ[redacted]大[redacted]は[redacted] 336
[redacted]は[redacted]ア[redacted]
[redacted]/[redacted]ア[redacted]大[redacted]に[redacted] 338
[redacted]お[redacted]/ア[redacted] TCP IP
[redacted]大[redacted]は[redacted]
[redacted]要[redacted]ア[redacted]
[redacted]大[redacted]は[redacted]18 14 29
[redacted]は[redacted]
[redacted]要[redacted]
[redacted]大[redacted]と[redacted]32[redacted]大[redacted]3[redacted]250
[redacted]大[redacted]と[redacted]お[redacted]

[redacted]5 386

[redacted] IP

[redacted]P

[redacted] 390

[redacted] 6 388

[redacted] 390

[redacted] 7 392

[redacted] 394

[redacted]0

[redacted] 410

[redacted] 412

[redacted] 414

[redacted] 414 2

[redacted] IP 420 TCP

[redacted]TCP IP[redacted] 3 424

[redacted] 426 4 428

[redacted] 426

[redacted] IP TDI

[redacted]IP[redacted] API

[redacted] IP

[redacted] TCP IP

[redacted] 5 430

[redacted] IP

[redacted]1

[redacted]IP[redacted] 21 Microsoft Window

[redacted]378 [redacted]PC

[redacted] 450

[redacted] WinSock

382 [redacted] TCP IP TCP

[redacted]4[redacted]

[redacted]38[redacted]

[redacted]82 IP

[redacted]CP 454 IP

[redacted] IP

[redacted] NDIS MA

[redacted]CP [redacted] 458

[redacted] IP

[redacted] IP

[redacted]M[redacted]
[redacted]の[redacted]459　　[redacted]IX Daemon486
[redacted]スと[redacted]　　UNIX
[redacted]は[redacted]459
[redacted]460[redacted]　　TCP　476 IP
[redacted]46と[redacted]P[redacted]　　488
[redacted]　　492
[redacted]1
[redacted]459[redacted]か[redacted]　　480　　I
[redacted]466[redacted]
[redacted]466[redacted]　　480
[redacted]　　1
[redacted]　　23
[redacted]Sock[redacted]　　247
[redacted]　　500
[redacted]CP　454[redacted]
[redacted]　　502
[redacted]
[redacted]2[redacted]
[redacted]　　DHCP
[redacted]
[redacted]2[redacted]　　DHCP
[redacted]4[redacted]
[redacted]TO[redacted]　　30 IP
[redacted]　　IP
[redacted]　　IP
[redacted]
[redacted]は[redacted]5 UNIX[redacted]TCP[redacted]
[redacted]と[redacted]IP[redacted]
[redacted]は[redacted]4[redacted]は[redacted]　　504
[redacted]
[redacted]
[redacted]4[redacted]　　508
[redacted]
[redacted]は[redacted]IP[redacted]
[redacted]　　TCP IP
[redacted]48と[redacted]P[redacted]
[redacted]　　510
[redacted]2
[redacted]
[redacted]　　I
[redacted]8[redacted]IX[redacted]
[redacted]　　TCP IP

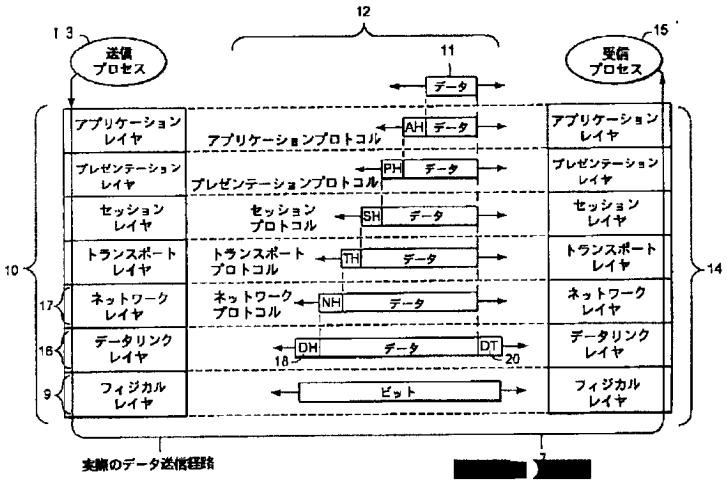
31 32

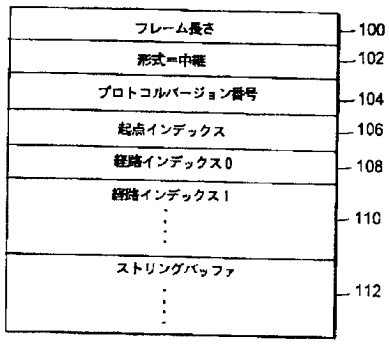
[redacted]

33　　34

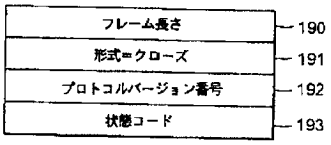
[redacted]　7
[redacted]　9
第[redacted]　10　1
[redacted]　11
[redacted]　12
送[redacted]ス　13
第[redacted]　14　2
[redacted]ス　15
[redacted]　16
[redacted]　17
[redacted]　18
[redacted]ラ　20
[redacted]　TC
[redacted]　P
[redacted]　26　IP
[redacted]　ARP
[redacted]
[redacted]ベ　46　A
[redacted]20　48　N1
第[redacted]　50　1
[redacted]　52
[redacted]　54　B
第[redacted]　58　2
[redacted]　60　N2
[redacted]ベ　62　D

[redacted]　1　[redacted]　6

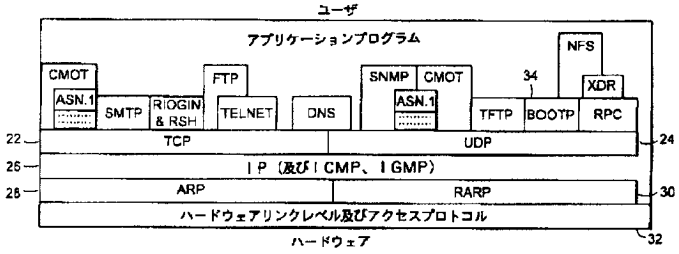




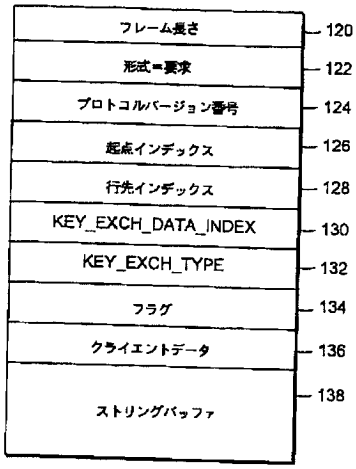
10



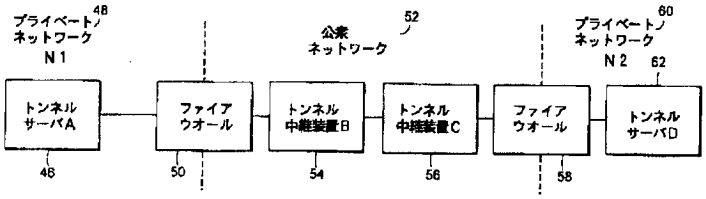
[redacted]　2



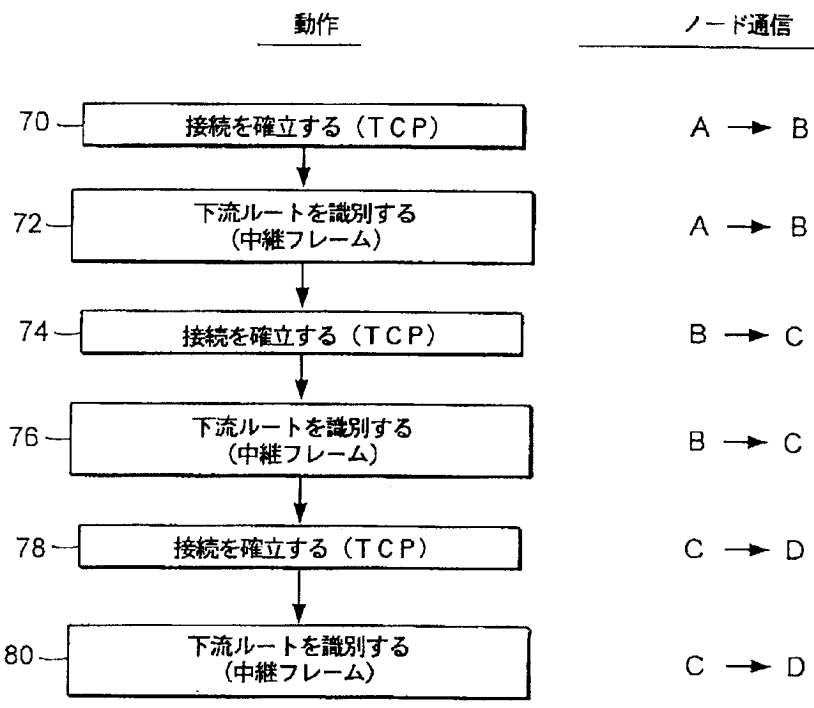
[redacted]　7



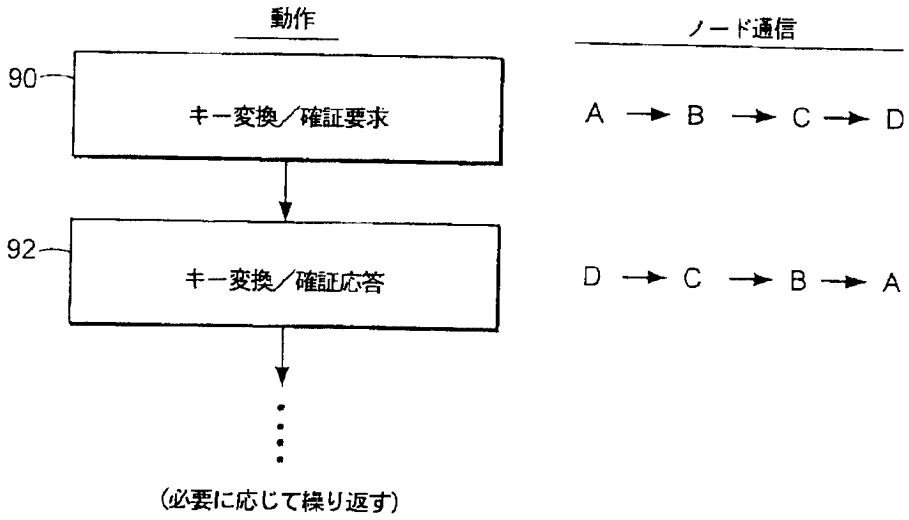
[redacted]　3



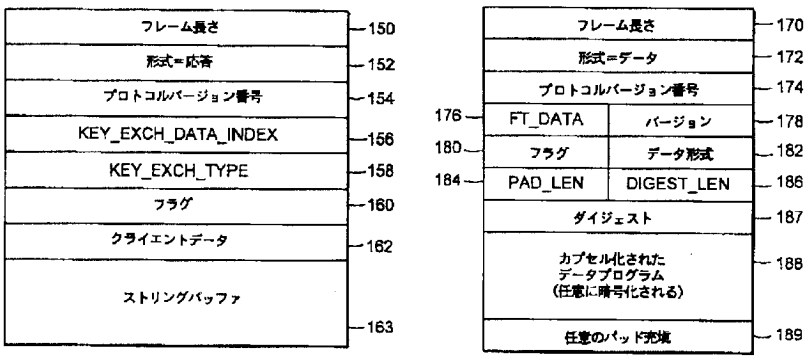
[redacted]　4



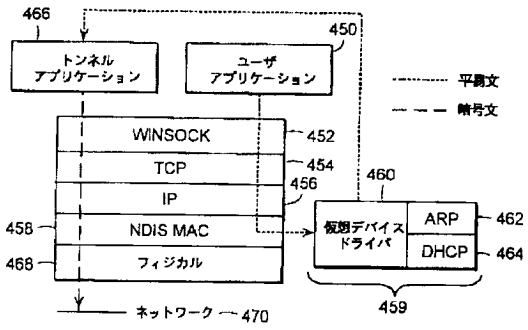
[redacted] 5



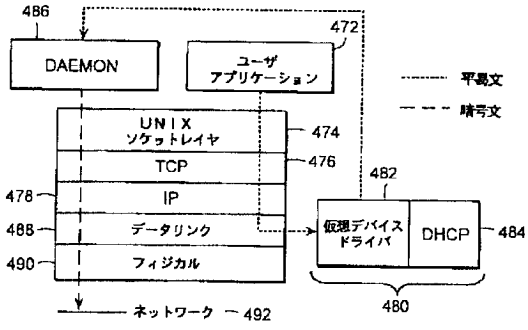
[redacted] [redacted] 9



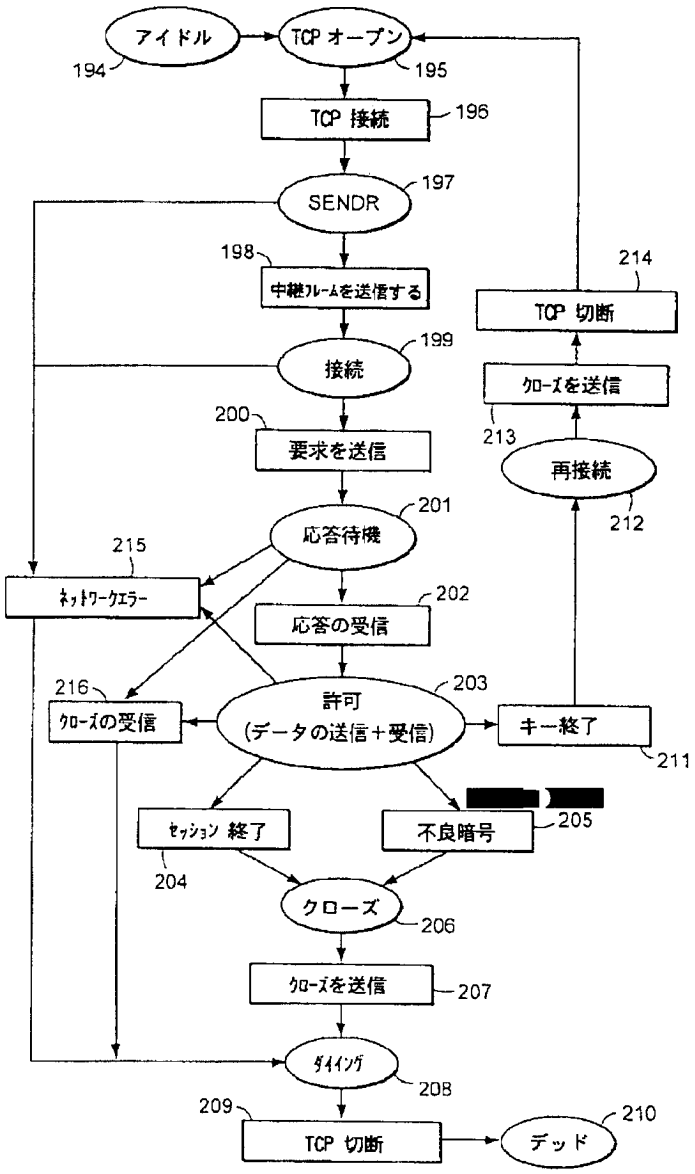
[redacted] [redacted] 21

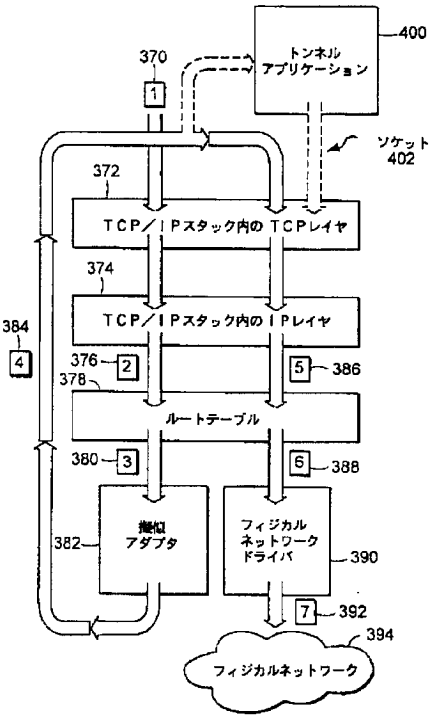


[redacted] [redacted] 22

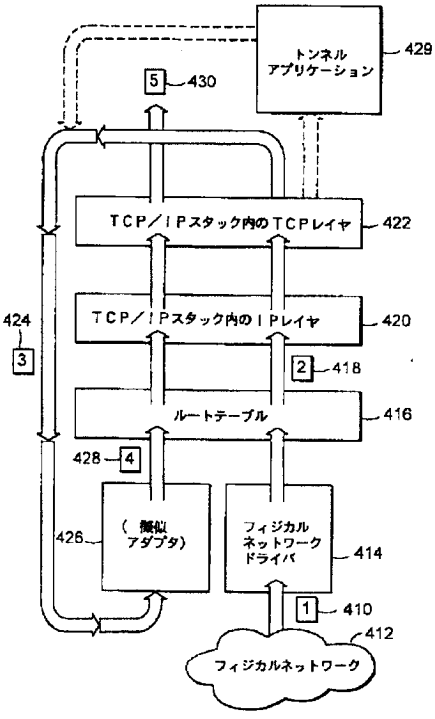


[redacted]　11　[redacted]　19

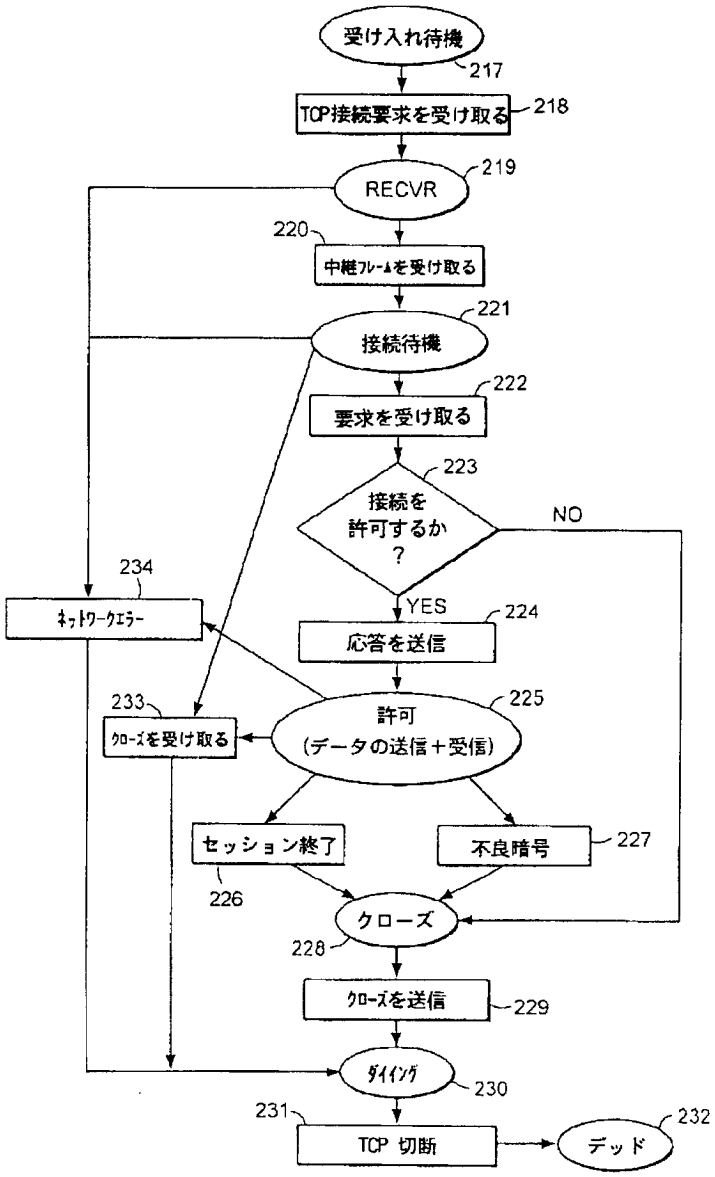




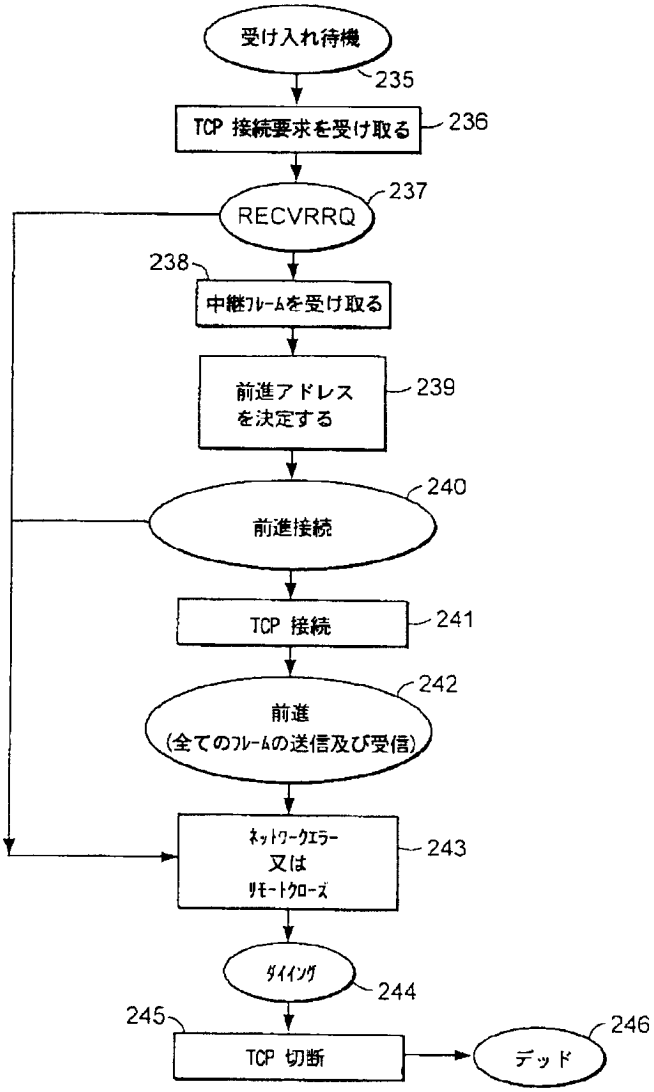
20

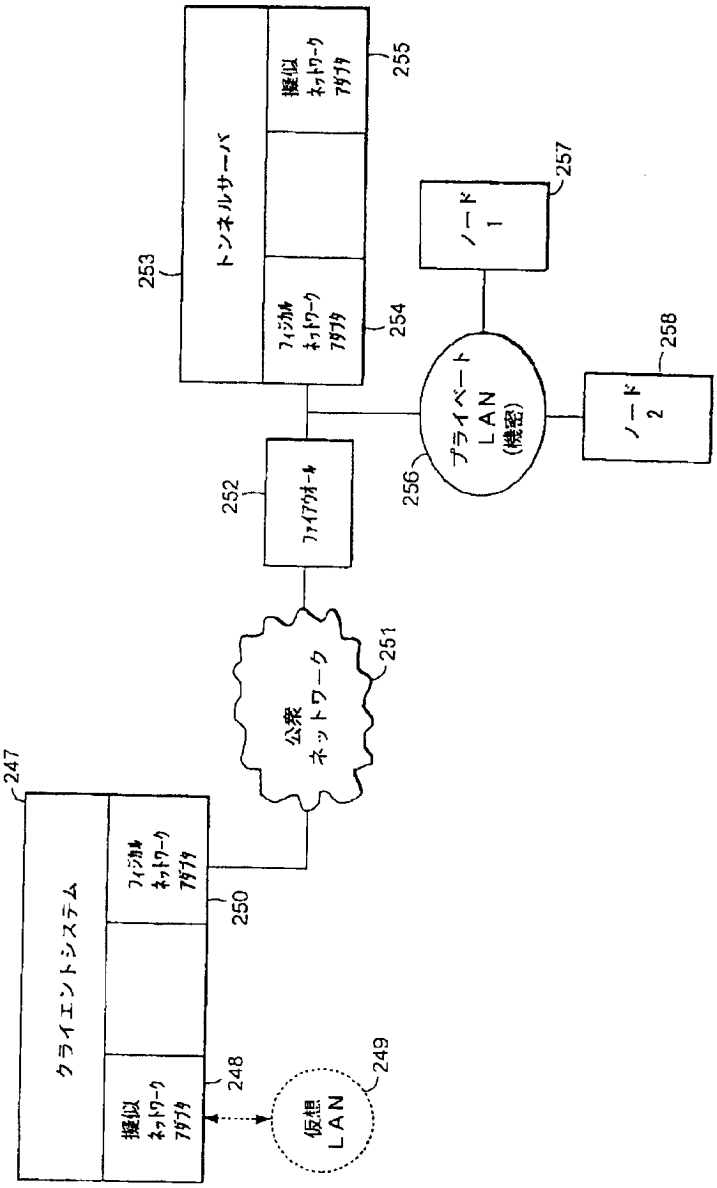


[redacted]　12

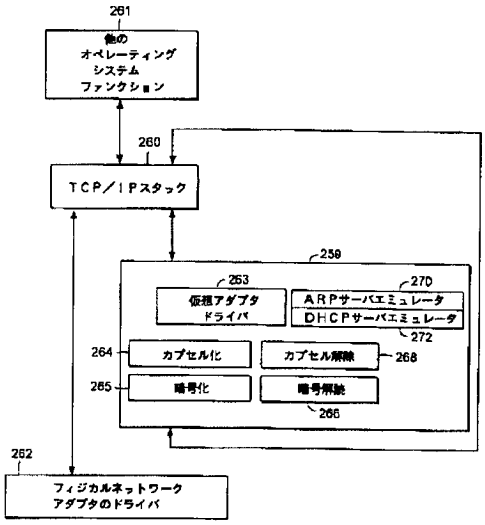

受け入れ待機
217
TCP接続要求を受け取る
218
RECVR
219
220
中継フレームを受け取る
接続待機
221
要求を受け取る
222
接続を許可するか？
223
NO
YES
234
ネットワークエラー
応答を送信
224
許可（データの送信＋受信）
225
233
クローズを受け取る
セッション終了
226
不良暗号
227
クローズ
228
クローズを送信
229
ダイイング
230
231
TCP 切断
232
デッド

[redacted] 13

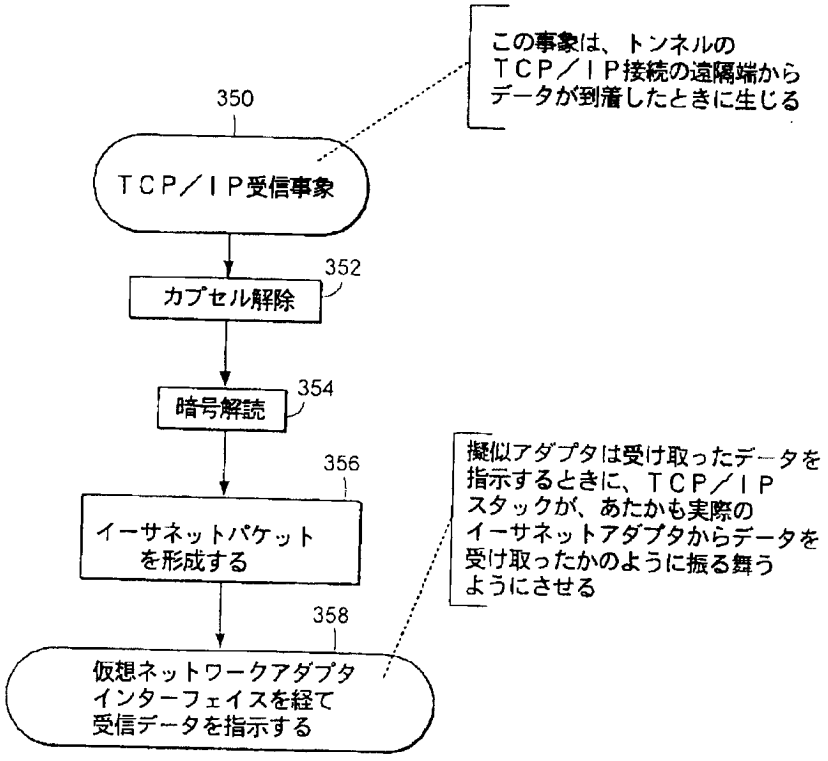

受け入れ待機
235
TCP 接続要求を受け取る
236
RECVRRQ
237
238
中継ﾌﾚｰﾑを受け取る
前進アドレス
を決定する
239
240
前進接続
TCP 接続
241
242
前進
(全てのﾌﾚｰﾑの送信及び受信)
ﾈｯﾄﾜｰｸｴﾗｰ
又は
ﾘﾓｰﾄｸﾛｰｽﾞ
243
ﾀｲｲﾝｸﾞ
244
245
TCP 切断
246
デッド

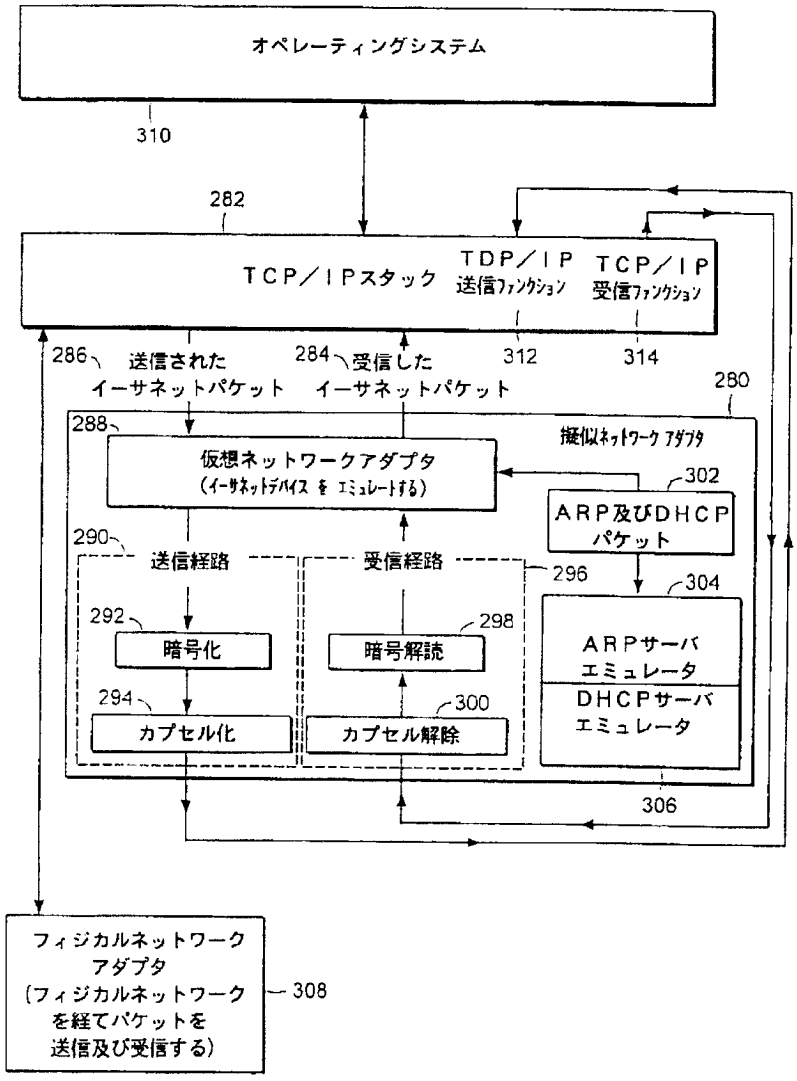

クライエントシステム
247
擬似ネットワークアダプタ
248
フィジカルネットワークアダプタ
250
仮想LAN
249
公衆ネットワーク
251
ファイアウォール
252
トンネルサーバ
253
擬似ネットワークアダプタ
255
フィジカルネットワークアダプタ
254
プライベートLAN(機密)
256
ノード1
257
ノード2
258

[redacted] 15

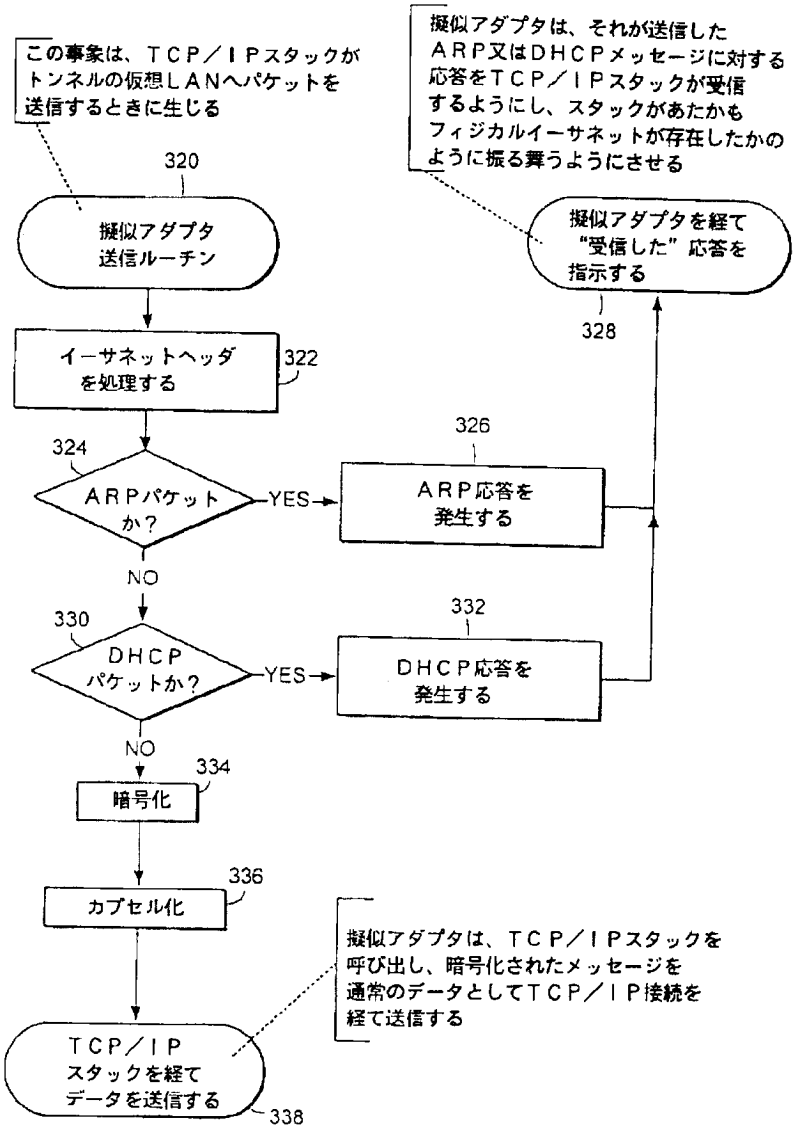


[redacted] 18

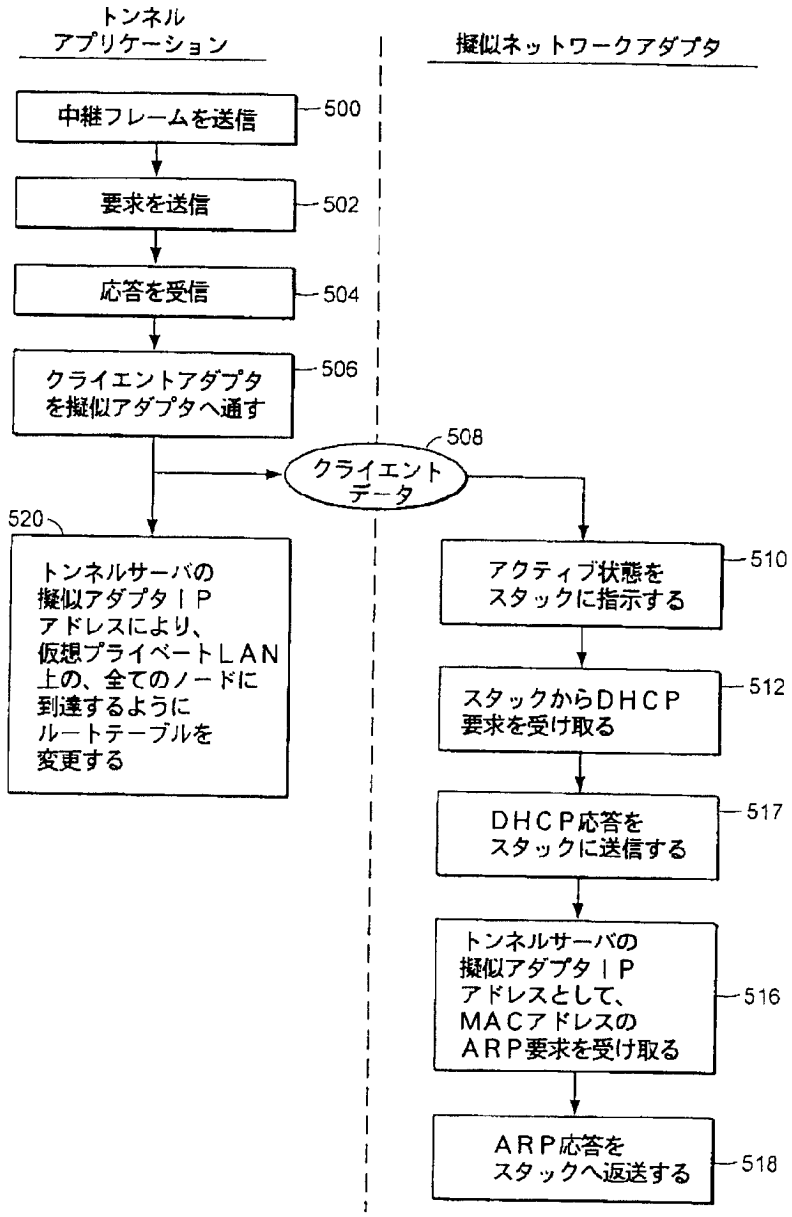



オペレーティングシステム
310
282
TCP／IPスタック
TDP／IP 送信ファンクション
TCP／IP 受信ファンクション
286
送信された イーサネットパケット
284
受信した イーサネットパケット
312
314
280
擬似ネットワーク アダプタ
288
仮想ネットワークアダプタ (イーサネットデバイス を エミュレートする)
302
ARP及びDHCP パケット
290
送信経路
受信経路
296
304
292
暗号化
298
暗号解読
ARPサーバ エミュレータ
294
カプセル化
300
カプセル解除
DHCPサーバ エミュレータ
306
フィジカルネットワーク アダプタ (フィジカルネットワークを経てパケットを送信及び受信する)
308

[redacted] 17


この事象は、ＴＣＰ／ＩＰスタックがトンネルの仮想ＬＡＮへパケットを送信するときに生じる
320
擬似アダプタ送信ルーチン
イーサネットヘッダを処理する
322
324
ＡＲＰパケットか？
YES
NO
326
ＡＲＰ応答を発生する
330
ＤＨＣＰパケットか？
YES
NO
332
ＤＨＣＰ応答を発生する
334
暗号化
336
カプセル化
ＴＣＰ／ＩＰスタックを経てデータを送信する
338
擬似アダプタは、ＴＣＰ／ＩＰスタックを呼び出し、暗号化されたメッセージを通常のデータとしてＴＣＰ／ＩＰ接続を経て送信する
擬似アダプタは、それが送信したＡＲＰ又はＤＨＣＰメッセージに対する応答をＴＣＰ／ＩＰスタックが受信するようにし、スタックがあたかもフィジカルイーサネットが存在したかのように振る舞うようにさせる
擬似アダプタを経て“受信した”応答を指示する
328

[redacted] 23



[redacted](51)Int.Cl.[6] F I
// H04L 9/36

[redacted]
[redacted]
[redacted]

[redacted]
[redacted]
[redacted]
[redacted]

PATENT ABSTRACTS OF JAPAN

(11)Publication number : **10-178450**
(43)Date of publication of application : **30.06.1998**

(51)Int.Cl. **H04L 12/56**
G09C 1/00
H04L 12/46
H04L 12/28
H04L 12/22
// H04L 9/36

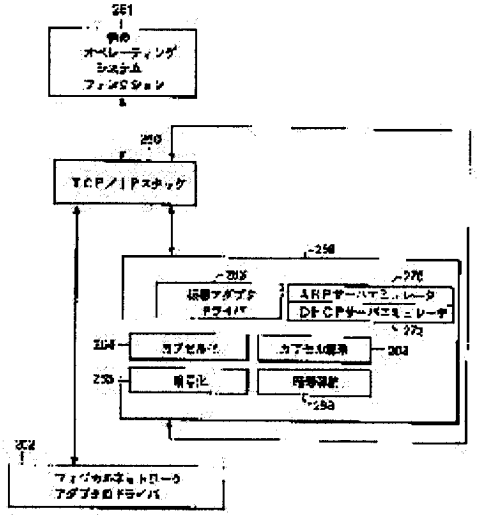
(21)Application number : **09-290739**
(22)Date of filing : **23.10.1997**

(71)Applicant : **DIGITAL EQUIP CORP <DEC>**
(72)Inventor : **ALDEN KENNETH F**
LICHTENBERG MITCHELL P
WOBBER EDWARD P

(30)Priority
Priority number : **96 738155** Priority date : **25.10.1996** Priority country : **US**

(54) PSEUDO NETWORK ADAPTOR FOR ACQUIRING, ENCAPSULATING AND ENCRYPTING FRAME

(57)Abstract:
PROBLEM TO BE SOLVED: To provide a pseudo network adaptor to acquire, encapsulate and encrypt a message or a frame.
SOLUTION: A pseudo network adaptor 259 is provided with a dynamic host configuration protocol(DHCP) server emulator 272 and an address resolution protocol(ARP). The system gives an instruction of it to a local communication protocol stack 260 that data reach a node of a remote private network via a gateway (GW) and the GW via the pseudo network adaptor. A data packet from the stack is processed and sent through a transmission path via the pseudo network adaptor. An encryption engine 265 encrypts the data packet and an encapsulate engine 264 encapsulates the encrypted data packet into a tunnel data frame. A transport layer of the stack to acquire the data packet received from a remote server node and a reception path processing the acquired data packet are provided.



* NOTICES *

CLAIMS

[Claim(s)]
[Claim 1]In a pseudo network adaptor provided with a virtual private network, From a local communication protocol stack, in order to transmit through the above-mentioned virtual private network, have an interface which captures a packet, and this interface, Are visible to the above-mentioned local communication protocol stack as a network adaptor device driver for a network adaptor connected to the above-mentioned virtual private network, It has the 1st server emulator which answers the 1st request packet captured with the above-mentioned interface which captures a packet from the above-mentioned local communication protocol stack in order to transmit through the above-mentioned virtual private network, and gives the 1st response packet, The 1st request packet of the above requires a network layer address for the above-mentioned pseudo network adaptor, and the 1st response packet of the above points to a network layer address for the above-mentioned pseudo network adaptor, and further, It has the 2nd server emulator which answers the 2nd request packet captured with the above-mentioned interface which captures a packet from the above-mentioned local communication protocol stack in order to transmit through the above-mentioned virtual private network, and gives the 2nd response packet, The 2nd request packet of the above requires a physical address corresponding to a network layer address of the 2nd pseudo network adaptor located in a remote server node, And a pseudo network adaptor, wherein the 2nd response packet of the above directs a physical address specified [predetermined].
[Claim 2]It has further a means to direct that it may arrive at a physical address specified [above-mentioned / predetermined] through the above-mentioned pseudo network adaptor to the above-mentioned local communication protocol stack, The pseudo network adaptor according to claim 1 which changes a data structure of the above-mentioned local communication protocol stack which directs which node or network this directing means may reach through each network interface of a local system.
[Claim 3]To one or more nodes of a remote private network connected to the above-mentioned remote server node. The pseudo network adaptor according to claim 1 further provided with a means to direct that it may reach through a gateway node equal to the 2nd pseudo network adaptor of the above of the above-mentioned remote server node to the above-mentioned local communication protocol stack.
[Claim 4]A transmission path for processing a data packet captured with the above-mentioned interface which captures a packet from the above-mentioned local communication protocol stack in order to transmit through the above-mentioned virtual private network, An encryption engine for being in the above-mentioned transmission path and enciphering the above-mentioned data packet, An encapsulation engine for being in the above-mentioned transmission path and encapsulating a data packet enciphered [above-mentioned] to a tunnel data frame, The pseudo network adaptor according to claim 1 further provided with a means to pass the above-mentioned tunnel data frame to the above-mentioned local communication protocol stack in order to transmit to a physical network adaptor of the above-mentioned remote server node.
[Claim 5]the above-mentioned transmission path equipping the digest field of each above-mentioned tunnel data frame with a means to memorize a digest value, further, and this digest value, It is the data packet encapsulated in the above-mentioned tunnel data frame, The pseudo network adaptor according to claim 4 equal to an output of a key type hash function applied to a data packet connected with a counter value equal to total of a tunnel data frame already transmitted to the above-mentioned remote server node.
[Claim 6]The pseudo network adaptor according to claim 4 including that the above-mentioned transmission path is provided with a means for processing further each Ethernet header of a data packet captured [above-mentioned], and the above-mentioned processing of the above-mentioned Ethernet header removes the above-mentioned Ethernet header.

[Claim 7]The pseudo network adaptor according to claim 1 further provided with an interface to a transport layer of the above-mentioned local communication protocol stack for capturing a data packet received from the above-mentioned remote server node.
[Claim 8]A received path for processing a received data packet which was captured with the above-mentioned interface to the above-mentioned transport layer of the above-mentioned local communication protocol stack for capturing a data packet received from the above-mentioned remote server node, The account of the upper by being in the above-mentioned received path and removing a tunnel frame header A capsule release engine for carrying out capsule release of the received data packet, The account of the upper with a decryption engine for deciphering a data packet which is in the above-mentioned received path and the account of the upper received in order to give to a user, a received data packet, The pseudo network adaptor according to claim 7 further provided with a means for returning the above-mentioned local communication protocol stack.
[Claim 9]In a method of forming a pseudo network adaptor for a virtual private network, From a local communication protocol stack, capture a packet, in order to transmit through the above-mentioned virtual private network, and this capture, It carries out through an interface which appears to the above-mentioned local communication protocol stack as a network adaptor device driver for a network adaptor connected to the above-mentioned virtual private network, Answer the 1st request packet captured with the above-mentioned interface which captures a packet from the above-mentioned local communication protocol stack in order to transmit through the above-mentioned virtual private network, and the 1st response packet is generated, Require the 1st request packet of the above and a network layer address for the above-mentioned pseudo network adaptor and the 1st response packet of the above, It points to a network layer address for the above-mentioned pseudo network adaptor, And answer the 2nd request packet captured with the above-mentioned interface which captures a packet from the above-mentioned local communication protocol stack in order to transmit through the above-mentioned virtual private network, and the 2nd response packet is generated, The 2nd request packet of the above requires a physical address corresponding to a network layer address of the 2nd pseudo network adaptor located in a remote server node, And a method, wherein the above-mentioned ARP response packet is provided with a stage of directing a physical address specified [predetermined].
[Claim 10]It has further a stage of directing that it may arrive at a physical address specified [above-mentioned / predetermined] through the above-mentioned pseudo network adaptor to the above-mentioned local communication protocol stack, A method according to claim 9 of changing a data structure of the above-mentioned local communication protocol stack which directs which node or network this stage directed to the above-mentioned local communication protocol stack may reach through each network interface of a local system.

[Claim 11]To one or more nodes of a remote private network connected to the above-mentioned remote server node. It has further a stage of directing that it may reach through a gateway node equal to the 2nd pseudo network adaptor of the above of the above-mentioned remote server node to the above-mentioned local communication protocol stack, To one or more nodes of a remote private network connected to the above-mentioned remote server node. The above-mentioned stage directed to the above-mentioned local communication protocol stack that it may reach through a gateway node equal to the 2nd pseudo network adaptor of the above of the above-mentioned remote server node, A method according to claim 9 of changing a network layer route table of the above-mentioned local communication protocol stack.
[Claim 12]A data packet captured with the above-mentioned interface which captures a packet from the above-mentioned local communication protocol stack in order to transmit through the above-mentioned virtual private network in a send data course is processed, In the above-mentioned transmission path, the above-mentioned data packet is enciphered with an encryption engine, A data packet enciphered [above-mentioned] with an encapsulation engine in the above-mentioned transmission path is encapsulated to a tunnel data frame, And in order to transmit the above-mentioned tunnel data frame to a physical network adaptor of the above-mentioned remote server node, Pass to the above-mentioned local communication protocol stack, and the above-mentioned transmission path, a digest value being memorized in the digest field of each above-mentioned tunnel data frame, and this digest value, It is the data packet encapsulated in the above-mentioned tunnel data frame, A method according to claim 9 equal to an output of a key type hash function applied to a data packet connected with a counter value equal to total of a tunnel data frame already transmitted to the above-mentioned remote server node.
[Claim 13]A method according to claim 12 of including that the above-mentioned processing of the above-mentioned Ethernet header removes the above-mentioned Ethernet header including the above-mentioned

transmission path processing further each Ethernet header of a data packet captured [above-mentioned].
[Claim 14]A stage of capturing a data packet received from the above-mentioned remote server node through an interface to a transport layer of the above-mentioned local communication protocol stack is included further, A received data packet which was captured with the above-mentioned interface to the above-mentioned transport layer of the above-mentioned local communication protocol stack for capturing a data packet received from the above-mentioned remote server node in a received path is processed, Capsule release of the data packet which the account of the upper received by removing a tunnel frame header with a capsule release engine in the above-mentioned received path is carried out, A method according to claim 9 which deciphered a data packet which the account of the upper received with a decryption engine in the above-mentioned received path, and was provided with a stage of returning a data packet which the account of the upper received to the above-mentioned local communication protocol stack in order to give to a user.
[Claim 15]A method according to claim 9 of communicating from the above-mentioned remote server node to the above-mentioned pseudo network adaptor as client data [in / in the above-mentioned network layer address of the above-mentioned pseudo network adaptor, and a physical address specified / predetermined / above-mentioned / a connection response frame].

[Translation done.]

TECHNICAL FIELD

[Field of the Invention]Generally, this invention relates to a secret virtual private network, and relates to the pseudo network adaptor for capturing and encapsulating a message or a frame in details, and enciphering in them more.

[Translation done.]

PRIOR ART

[Description of the Prior Art]In data communications, to give secret communication among the users of a network station (it is also called a network node) in a different physical location is often needed. Secret communication must be potentially prolonged through a secret private network through a public network. A secret private network is protected by "the firewall (fire wall)" which separates the private network from a public network. A firewall gives the combination which generally has a packet filter for insulating a private network from the undesirable communication with a public network, a circuit gateway, and application gateway art.
[0003]One solution for giving secret communication is forming a virtual private network. In a virtual private network, secret communication is given by encapsulating and enciphering a message. Generally the encapsulated message transmission is called "tunnel transmission." The tunnel using encryption gives the communication protected among the users separated by the public network or between the subsets of the user of a private network.
[0004]Encryption is performed using the encryption algorithm which uses one or more codes "key", for example. When a cryptographic key is used, it is determined how the value of a key enciphers and decodes data. When a public key encryption system is used, a key pair relates to each communication entity. A key pair comprises a cryptographic key and a decode key. These two keys are made so that one key cannot be formed from the key of another side. Although each entity exhibits the cryptographic key, it maintains the decode key at secrecy. For example, when transmitting a message to the node A, the transmitting side entity cannot encipher a message using the public key of the node A, and the message can decode only the node A using the private key of the node A.
[0005]In a symmetrical key encryption system, a single key is used as a base to both a code and a decipherment. The cryptographic key in a symmetrical key encryption system may be called a "common" key. For example, the communication nodes A and B of a couple can communicate in secrecy as follows. That is, while enciphering the data sent to the node B from the node A using the 1st common key, the 2nd common key is used for enciphering the data sent to the node A from the node B. In such a system, both the node A and the node B must know two common keys. Another example of encryption algorithm and key type encryption Much articles. For example, "protocol [in application cryptography-C], algorithm, and source code (Applied Cryptography - Protocols, Algorithms and Source Code in C)" blues SHUNAIRA work, It is indicated in John wheelie and SONZU publication of New York State and New York, and copyright 1994.
[0006]The information about what kind of cryptographic key should be used and how the data of a given secret communication session should be enciphered using them is called "key exchange." What kind of key is used or key exchange determines time width with each effective key, for example. Before the key exchange to the communication stations of a couple exchanges encryption data in a secret communication session, both stations must know it. It is called "session key establishment" how communication stations make key exchange known to given secret communication.
[0007]It is visible as a physical device to a communication protocol stack, and can tunnel using the imagination, i.e., the "false" network adaptor, which form a virtual private network. It must have the capability to transmit.
[0008]The end point of a tunnel is a point on which the code / decipherment, and the encapsulation / capsule release given by tunnel is performed. In the existing system, the end point of a tunnel is the

network layer (layer) address determined beforehand. "Credibility" of the entity which requires establishment of tunnel connection is determined using the sauce network layer dress in the received message. For example, a tunnel server determines whether the demanded tunnel connection is permitted using a sauce network layer dress. A sauce network layer dress is used also for determining which cryptographic key should be used for decoding the received message.

[Translation done.]

* NOTICES *

JPO and INPIT are not responsible for any damages caused by the use of this translation.

1.This document has been translated by computer. So the translation may not reflect the original precisely.
2.**** shows the word which can not be translated.
3.In the drawings, any words are not translated.

TECHNICAL PROBLEM

[Problem(s) to be Solved by the Invention]The existing tunnel transmitting art is performed by generally encapsulating the enciphered network layer packet (it is also called a frame) in a network layer. Such a system gives "network layer in network layer" encapsulation of the enciphered message. The tunnel of the existing system exists between the firewall nodes which have the IP address usually assigned statically. The IP address to which the firewall was statically assigned in such an existing system is an address of the tunnel end point in a firewall. The existing system does not give the tunnel which can issue permission by using as a base the entity which must assign a network layer address dynamically. This poses a problem, when a user is going to establish a tunnel in move calculation environment especially, and when requiring the IP address dynamically assigned from the Internet Service Provider (ISP).
[0010]Since the existing virtual private network is using the network layer encapsulation in a network layer as the base, generally it can provide only datagram type service of a no connection. Since datagram type service does not guarantee supply of a packet, the existing tunnel can only use an encryption method easily to the data contained in the each transmitted packet. The encryption which uses the contents of many packets as a base, for example, the encryption block chain to many packets, and stream cipher-ization are desired. For example, as for encryption data, it is convenient not only based on the contents of the present packet data enciphered but to be formed also based on an attribute with the connection between communication stations or the career of a session. The example of encryption algorithm and key type encryption Much articles. For example, "protocol [in application cryptography-C], algorithm, and source code (Applied Cryptography - Protocols, Algorithms and Source Code in C)" blues SHUNAIRA work, It is indicated in John wheelie and SONZU publication of New York State and New York, and copyright 1994.
[0011]Therefore, in order to support the user who is in move calculation environment, the new pseudo network adaptor which gives the virtual private network which has the end point determined dynamically is demanded. This new pseudo network adaptor must appear as an interface to a actual physical device to the communication protocol stack of a node. This new pseudo network adaptor must support the orderly supply the frame passing through a tunnel was guaranteed to be, in order to support conveniently the encryption block chain mode or stream-cipher-izing to many packets.

[Translation done.]

* NOTICES *

MEANS

[Means for Solving the Problem]This invention provides a new pseudo network adaptor which gives a virtual private network. This new system is provided with an interface which captures a packet from a local communication protocol stack in order to transmit through a virtual private network. This interface is visible to a local communications stack as a network adaptor device driver for a network adaptor.
[0013]This invention includes the pseudo network adaptor according to claim 1 which gives a virtual network, and a method according to claim 9 in the large gestalt.
[0014]A system of this invention is further provided with a dynamic host configuration protocol (DHCP) server emulator and an address resolution protocol (ARP) server emulator so that it may state below. This new system can reach a node of a remote private network through a gateway, and directs that this gateway can be reached through a pseudo network adaptor to a local communication protocol stack. This new pseudo network adaptor is provided with a transmission line which processes a data packet from a local communication protocol stack in order to transmit through this pseudo network adaptor. This transmission line is provided with the following.
An encryption engine for enciphering a data packet.
An encapsulation engine for encapsulating an enciphered data packet to a tunnel data frame.
A pseudo network adaptor is returned to a local communication protocol stack in order to transmit a tunnel data frame to a physical network adaptor of a remote server node.
[0015]Preferably, a pseudo network adaptor contains a digest value in the digest field of each tunnel data frame so that it may state below. A key type hash function is a hash function which takes up data and a common cryptographic key as an input, and outputs digital numerals called a digest. A value of the digest field is equal to an output of a key type hash function, and this key type hash function, It is data which comprises a data packet encapsulated in a tunnel data frame, and is applied to data connected with a counter value equal to total of a tunnel data frame already sent to a remote server node. .In another feature of this system, a pseudo network adaptor processes an Ethernet header in each of a captured data packet, and this includes removing an Ethernet header.
[0016]A new pseudo network adaptor is provided with the following.
An interface to a conveyance layer of a local communication protocol stack for capturing a data packet received from a remote server node.
A received path for processing a received data packet which was captured from a conveyance layer of a local communication protocol stack.
A received path is returned to a local communication protocol stack in order to give to a user a data packet by which was provided with a capsule release engine and a decryption engine, and capsule release was deciphered and carried out.
[0017]Therefore, in order to support a user who is in move calculation environment, a new pseudo network adaptor which gives a virtual private network which has the end point determined dynamically is provided. This new pseudo network adaptor is provided with a system captured before transmitting a frame formed thoroughly. This new pseudo network adaptor appears as an interface to a actual physical device to a communication protocol stack of a station. This new pseudo network adaptor is provided also with an enciphering function for giving secret communication conveniently between end points of a tunnel using stream mode encryption or a code block chain to many packets.

[0018]
[Embodiment of the Invention]This invention will fully be understood from detailed explanation of the following of the desirable embodiment which referred to the accompanying drawing. First, with reference to drawing 1, the communication which uses an open system interconnection (OSI) reference model as a base is explained. The communication 12 between the 1st protocol stack 10 and the 2nd protocol stack 14 is shown in drawing 1. The 1st protocol stack 10 and the 2nd protocol stack 14, Seven protocol layers (an application layer, a presentation layer, a session layer, a transport layer, a network layer, a data link layer, and a physical layer) of an OSI reference model are carried out. In combination with software and hardware, a protocol stack is usually carried out. Explanation of the specific service given by each protocol layer of an OSI reference model, It will see in the Prentice Hall publication of much articles, for example, "computer network (Computer Networks)", the 2nd edition, undrew one, S and the Tanenbaum work, New Jersey, and Inglewood Cliff, and copyright 1988.

[0019]As shown in drawing 1, it lets caudad the data 11 which should be sent to the receiving process 15 from the transmitting process 13 pass through the protocol stack 10 of a transmitting process to the physical layer 9, and it is sent to the receiving process 15 through the data path 7. A header part is attached in order that each protocol layer may convey the information used by the protocol layer, when letting the data 11 pass caudad through the protocol stack 10 (and probably also in case of ** trailer portion). For example, the data link layer 16 of a transmitting process wraps the information received from the network layer 17 in the data link header 18 and the data link layer trailer 20, after that, it lets a message pass to the physical layer 9, and it is transmitted through the actual transmission path 7.
[0020]Drawing 2 shows a TCP/IP protocol stack. Some protocol layers of this TCP/IP protocol stack correspond to the layer of the OSI protocol stack shown in drawing 1. The service and the header format of each layer in a TCP/IP protocol stack, Much articles, for example, interworking with "TCP/IP, the 1st volume, A principle, a protocol, and the architecture" (Interworking with TCP/IP, Vol. 1: Principles, Protocols, and Architecture), It is explained to the Prentice Hall publication of the 2nd edition, the Douglas E. Commer work, New Jersey, and Inglewood Cliff, and the 1991 copyright in detail. The transport control protocol (TCP) 22 is equivalent to the transport layer of an OSI reference model. The TCP protocol 22 provides connection-oriented the end / end transport service accompanied by the guaranteed order right packet supply. Thus, the TCP protocol 22 gives reliable transport layer connection.
[0021]IP protocol 26 is equivalent to the network layer of an OSI reference model. IP protocol 26 does not give a guarantee at all about the packet supply to a user layer. A hardware link level and the access protocol 32 are equivalent to the data link and physical layer of an OSI reference model.
[0022]Address Resolution Protocol (ARP) 28 is used for carrying out the map of the IP layer address (an IP address is called) to a hardware link level and the access protocol 32 (a physical address or a MAC Address is called). The ARP protocol layer of each network station is usually provided with the table (ARP cash is called) of mapping between an IP address and a physical address. When mapping between an IP address and the physical address corresponding to it is strange, the ARP protocol 28 generates a broadcasting packet (ARP request packet) in a local network. An ARP demand directs the IP address as which a physical address is required. The ARP protocol 28 of each station connected to the local network, When a station checks the IP address which considered the ARP demand and was directed by the ARP demand, the response (an ARP response or an ARP rep rye packet) for which a responder's physical address is directed is generated to the station which has emitted the demand. The ARP protocol which has emitted the demand reports the received physical address to a local IP layer, and this local IP layer ranks second and transmits datagram to a responder station directly using it. Use an ARP server, it is made to answer 1 set of IP addresses which it memorizes inside apart from each station answering only the IP address of itself, and, thereby, the necessity for a broadcasting demand can also be eliminated potentially. In this case, an ARP demand can be directly transmitted to an ARP server to the physical address corresponding to IP address mapping memorized in the ARP server.
[0023]Before each station on a network communicates using TCP/IP at the time of start up of a system, it must opt for the IP address to each of the network interface. For example, the station

must take a server and contact, in order to obtain dynamically the IP address to one or more network interfaces. What is called a dynamic host configuration protocol (DHCP) can be used for a station, and it can generate the demand of an IP address to a DHCP server. For example, a DHCP module transmits the DHCP request packet which requires the assignment of an IP address to the directed network interface at the time of start up of a system. A DHCP server will assign an IP address to the requestor-side station used with the directed network interface, if a DHCP request packet is received. Subsequently, the requestor-side station is memorized from a server to a response as an IP address for relating the IP address with the network interface, when communicating using TCP/IP.
[0024]Drawing 3 shows the example of composition of the network node which can apply the system indicated here. In the example of drawing 3, the tunnel server A is a start side of tunnel connection. As shown in drawing 3, the term of a "tunnel relay" node is used as what points out the station which transmits a data packet between transport layer connection (for example, TCP connection).
[0025]For example, in the system of this invention, tunnel repeating installation can be dynamically constituted so that a packet may be transmitted between the transport layer connections 2 with the transport layer connection 1. Tunnel repeating installation replaces the header information of the packet received through the transport layer connection 1 with the header information which shows the transport layer connection 2. Subsequently, the tunnel repeating installation can transmit a packet to a firewall, and this firewall can be programmed conveniently to let the packet received through the transport layer connection 2 pass to the private network by the side of other [of a firewall]. In the system of this invention, tunnel repeating installation forms transport layer connection dynamically, when tunnel connection is established. Therefore, when tunnel connection is established, the tunnel repeating installation can maintain dynamic load balance to one or more tunnel services, or can give the redundant service for defective permission.
[0026]Drawing 3 is shown where the tunnel server A (46) of the private network N1 (48) is physically connected to the 1st firewall 50. The 1st firewall 50 separates the private network N1 (48), for example from the public network 52 like the Internet. The 1st firewall 50 is physically connected to the tunnel repeating installation B (54), for example, and this repeating installation ranks second and is physically connected with the tunnel repeating installation C through the public network 52. The connection between the tunnel repeating installation B and the tunnel repeating installation C straddles the transmission node between which it is placed through the public network 52 by a large number like a router or a gateway, for example.
[0027]The tunnel repeating installation C is physically connected to the 2nd firewall 58, and this firewall separates the public network 52 from the private network N2 (60). The 2nd firewall 58 is physically connected with the tunnel server D (62) of the private network N2 (60). The tunnel server D (62) gives the route of an IP packet working [the element shown in drawing 3] between the tunnel connection with tunnel server A (46), and other stations on the private network N2 (60). Thus, the tunnel server D (62) works as tunnel connection and a router between the private networks N2 (60).
[0028]The system of this invention establishes tunnel connection working [the element shown in drawing 3] between the private network N1 (48) and the private network N2 (60). Therefore, the embodiment of drawing 3 eliminates the physical cable for exclusive use for giving secret communication between the private network 48 and the private network 60, or the necessity for a line. The tunnel connection between tunnel server A (46) and tunnel server D (62), It comprises transport layer connection of the positive opposite method between tunnel repeating-installation C (56), the (node C), and tunnel server D (62) and the (node D). [tunnel server A (46), the (node A), tunnel repeating-installation B (54) and the (node B), and] For example, connection of such an opposite method is each transport layer connection between each node A and the node B, between the node B and the node C, and between the node C and the node D. According to another embodiment, tunnel connection may be connected between stand-alone PC of a public network, and the tunnel server in a private network so that it may state below.
[0029]Drawing 4 and 5 illustrate the stage performed while establishing tunnel connection between tunnel server A (46), the (node A), and tunnel server D (62) and the (node D), as shown in drawing 3. Before the stage shown in drawing 4, the node A chooses the tunnel course for reaching the node D. A tunnel course contains the end point and the intervening tunnel repeating installation of a tunnel. A tunnel course is beforehand determined by the system management equipment of the node A, for

example. Each tunnel repeating installation in alignment with a tunnel course, For example, the next node of a tunnel course can be found using the domain name system (DNS) of the naming regulation beforehand determined based on the given following node name (or next nodearc), and service, for example, a TCP/IP protocol suite.

[0030]In the stage shown in drawing 4, each of the node A, B, and C performs the following stage.

- For example, analyze the node name of the next node of the tunnel course found in the tunnel relay frame.
- Establish the positive transport layer (TCP) connection with the next node of a tunnel course.
- Transmit a tunnel relay frame to the next node of a tunnel course along with the newly formed positive transport layer connection.

[0031]For example, as shown in drawing 4, in Step 70, the node A establishes the positive transport layer connection with the node B. In Step 72, the node A identifies the following downstream node to the node B by sending a tunnel relay frame to the node B through the positive transport layer connection between the node A and the node B. A tunnel relay frame contains the string buffer which describes all the nodes in alignment with a tunnel course (see the explanation to the following of an illustration tunnel relay frame format). In Step 74, it is determined whether answer the tunnel relay frame from the node A, the node B searchs the string buffer of a relay frame, and a string buffer includes the node name of the node B. If the node B finds the node name in a string buffer, the following node name will be looked for in a string buffer, and the node name of the next node of a tunnel course will be found.

[0032]The node B establishes the positive transport layer connection with the next node C of a tunnel course, for example, a node. The transport layer connection in which the node B is [between the node A and the node B from which the relay frame was received further] positive, The succession packet which formed the relation between the newly between the node B and the node C formed positive transport layer connection, and was received through the positive transport layer connection with the node A as a result is transmitted also to the positive transport layer connection with the node C, and its reverse. In Step 76, the node B transmits a tunnel relay frame to the node C through the newly formed positive transport layer connection.

[0033]In Step 78, the relay frame sent from the node B is answered, and it is determined that the next node of a tunnel course is a node of the last of a tunnel course, therefore the node C is a tunnel server. It can be determined actively whether tunnel server another in order to form tunnel connection can use the node C. The node C can choose one of the another tunnel servers which can be used, and can form the tunnel connection for giving load balance or defective permission. As the result, the node C can form the transport layer connection with the low tunnel server of availability comparatively, when established, one, for example, tunnel connection, of many usable tunnel servers. According to the embodiment illustrated here, the node C establishes the positive transport layer connection with the node D in the following node in alignment with a tunnel course, and this case.

[0034]The transport layer connection in which the node C is [between the node B and the node C from which the relay frame was received] positive, Form the relation between the newly between the node C and the node D formed positive transport layer connection, and as the result, The succession packet received through the positive transport layer connection with the node B is transmitted also to the positive transport layer connection with the node D, and its reverse. In Step 80, the node C transmits a relay frame to the node D through the newly formed positive transport layer connection.

[0035]Drawing 5 shows the common example of the key exchange given by the corroboration of a tunnel end point, and the system of this invention. The system of this invention supports letting corroboration data and key exchange pass to the positive transport layer connection beforehand established in the tunnel course. The following thing is given by use of key exchange / corroboration demand (REQUEST) frame and key exchange / corroboration response (RESPONSE) frame.

a) Mutual corroboration of both the end points of tunnel connection;

b) Establishment of the common session cryptographic key for enciphering / corroborating the succession data sent through tunnel connection, and a key life;

c) Exchange of data peculiar to other connection between tunnel end points of compression of agreement; about the common set of the encryption conversion which should be applied to succession data, d, for example, the strength of the code which should be used, and form, and the

data that should be used, etc. This data can also be used by the client of this protocol, in order to define the characteristic of the corroborated connection.

[0036]In Step 90, key exchange / corroboration demand frame is transmitted through the positive transport layer connection formed in the node D from the node A in accordance with the tunnel course. In Step 92, key exchange / corroboration response frame is returned to the node A from the node D through positive transport layer connection. The attribute exchanged using the stage shown in drawing 5 can be used among the life of tunnel connection. In another embodiment, the stage shown in drawing 5 The width of a cryptographic key and a key, And in order that the end point of a tunnel may exchange sufficient key exchange to agree based on 1 set of session parameters used during tunnel connection called selection of a code / decipherment algorithm, it is repeated if needed.

[0037]In the system indicated here, the name for using it for the corroboration and access control to the node A and the node D does not need to be a network layer address or a physical address of a node. For example, at another embodiment which is stand-alone type PC in which the start node which transmits a tunnel relay frame is located in a public network, a user's name can be used for the purpose of a corroboration and/or access control. This gives the remarkable improvement which excels the existing system which uses the corroboration based on a predetermined IP address as a base.

[0038]Drawing 6 shows the format of the embodiment of a tunnel relay frame. Drawing 6 and the tunnel frame format shown in 7, 8, and 9 are encapsulated in the data part of a transport layer (TCP) frame at the time of transmission. Or another equivalent connection-oriented transport layer protocol which has the guaranteed order right frame supply again may be used. A TCP frame format including a TCP header field is the usual thing, and is not illustrated.

[0039]The field 100 contains the length of a frame. The field 102 includes the form of a frame, for example, the form of RELAY (relay). The field 104 contains the version number of a tunnel protocol. The field 106 contains the index to the string buffer field 112 in which the name of a dispatch node, for example, the DNS host name of the node which generates a relay frame first, (the node A of drawing 3) is located. The field following this starting point index field 106 contains the index to the string buffer 112 with which the name of the node in alignment with a tunnel course is located. For example, each index may be offset of the DNS host name in the string buffer 112. Thus, the field 108 contains the index of the name of the 1st node B in a tunnel course, for example, a node, (drawing 3). The field 110 serves as **, such as an implication, in the index of the name of the 2nd node in a tunnel course. The field 112 contains the string of the node name of the node in a tunnel course.

[0040]The start node A, for example, the node shown in drawing 3, transmits a tunnel relay frame like the tunnel relay frame shown in drawing 6 working [the system of this invention]. The node A transmits a tunnel relay frame to the 1st station B in alignment with a tunnel course, for example, a node, (drawing 3) through the already established positive transport layer connection. The node B searchs the string buffer of a tunnel relay frame, and finds the node name, for example, the DNS host name. The node B finds the node name of the string buffer directed by the course index 0, and uses the contents of the course index 1 (110), and determines the position in the string buffer 112 of the node name of the following node in alignment with a tunnel course. The node B uses this node name and establishes the positive transport layer connection with the following node in alignment with a tunnel course. Subsequently, the node B transmits a relay frame to the following node. This process is continued until it reaches the last node of a tunnel route, for example, tunnel server D (62) and (drawing 3).

[0041]Drawing 7 shows the format of the embodiment of key exchange / key corroboration demand frame. The field 120 contains the length of a frame. The field 122 contains REQUEST (demand) which directs the form of a frame, for example, key exchange / key corroboration demand frame. The field 124 contains a tunnel protocol version number. The field 126 includes offset of the name of the entity which starts tunnel connection, for example, the name of the user of the node which generates a demand frame first. This name and key exchange in a demand frame are used for corroborating key exchange / corroboration demand by the end point of a receiver tunnel. The name of the entity which starts tunnel connection is used in order to permit following tunnel connection based on the secret predetermined plan of a system. The field 128 includes the offset to the frame of the node name of a destination node, for example, the terminal node of the tunnel shown in drawing 3 as node D (62).

[0042]The field 130 includes the offset to the frame key switched data is remembered to be in the string buffer field 138, for example. Key switched data includes the key exchange used for, for example, determining the common set of the cryptographic parameter to the life of tunnel connection like the earned hours relevant to a cryptographic key and these keys. The information about the strength of the common set of the cipher converting which should be used, and a parameter peculiar to other connection, for example, the code which should be used, and form, the form of compression of the data which should be used, etc. also includes key switched data and the field 132 further. Furthermore the field 134 is related with a frame, it contains the flag which directs another information, for example. Since a local route table is constituted so that the packet for the node which may reach through a virtual private network may be transmitted through a pseudo network adaptor, the field 136 contains the client data used for the end point of a tunnel. According to a certain embodiment, the string buffer 138 is enciphered using the open cryptographic key of the end point of a receiver tunnel.

[0043]Working [the system of this invention], one side of the terminal node of a tunnel transmits key exchange / corroboration demand frame shown in drawing 7 to the terminal node of another side of a tunnel, and performs the key exchange and the corroboration which were described in the stage 90 of drawing 5.

[0044]Drawing 8 shows the format of the embodiment of the key exchange / key corroboration response frame called a connection RESPONSE (response) frame. The field 150 contains the length of a frame. The field 152 includes the connection RESPONSE (response) which directs the form of a frame, for example, key exchange / key corroboration response frame. The field 154 contains a tunnel protocol version number.

[0045]The field 156 includes the offset to the frame key switched data is remembered to be for example, in the string buffer field 163. Key switched data includes the earned hours relevant to the key exchange which should be used for a code/decipherment over the life of tunnel connection, and this key exchange, for example. The information about compression etc. of the strength of the common set of the cipher converting which should be applied to succession data, and a parameter peculiar to other connection, for example, the code which should be used, and form, and the data that should be used also includes key switched data and the field 158 further. The field 160 contains the flag which directs other information about a frame, for example. The client data field 162, Since a local route table is constituted so that the packet for the node of a virtual private network may be transmitted through a pseudo network adaptor, the data used by the pseudo network adaptor of the end point of a tunnel is included. A string buffer includes key exchange. The tunnel end point of a receiver is used for a string buffer, the open cryptographic key of the opener of tunnel connection is used in this case, for example, and it is enciphered.

[0046]Working [the system of this invention], one side of the terminal node of a tunnel transmits key exchange / corroboration response frame shown in drawing 7 to the terminal node of another side of a tunnel, and performs the key exchange and the corroboration which were described in the stage 92 of drawing 5.

[0047]The format of the embodiment of the tunnel data frame used for drawing 9 communicating through tunnel connection is shown. Drawing 9 shows how an IP datagram is encapsulated in a tunnel frame by the system of this invention for the secret communication which passes along a virtual private network. The field 170 contains the length of a frame. The field 174 containing DATA (data) the field 172 instructs to be, the form, for example, the tunnel data frame, of a frame, contains the version number of a tunnel protocol.

[0048]The fields 176, 178, and 182 include the information about the encapsulated datagram. The field 180 contains the flag which directs the information about a frame. The field 184 contains the value which directs the length of the arbitrary pads 189 in the end of a frame. This frame format enables it to perform arbitrary pad restoration, when the frame data of the quantity which is in an even number block border in order to encipher using a block cipher needs to be stuffed. The field 186 contains the value which directs the length of the digest field 187.

[0049]This data frame format contains the digital numerals generated by the tunnel end point of the transmitting side called a "digest." The value of this digest secures data integrity, for example by detecting the response of an invalid frame and the already transmitted effective frame. A digest is an

output of the conventional key type encryption hash function applied to both the encapsulated datagram 190 and the sequence number which increases in monotone. It lets the hash output obtained by this pass as a value of the digest field 187. A sequence number is not included in a data frame.In this embodiment, a sequence number is a counter maintained by the transmitting side (for example, the node A of drawing 3) about all the data frames sent to the reception side node (for example, the node D of drawing 3) since establishment of tunnel connection.

[0050]In order to judge whether a data frame is invalid or it is a duplicate, a reception side node, Decipher the encapsulated datagram 190 and a key type encryption hash function (it has agreed by the terminal node of a tunnel into the stage shown in drawing 5) The deciphered encapsulation datagram, It applies to both values of the counter which directs the number of the data frames received from the transmitting side since establishment of tunnel connection. For example, a key type hash function is applied to the datagram connected with the counter value. When the hash output obtained by this is in agreement with the value of the digest field 187, the encapsulated datagram 190 is received correctly and is not a duplicate. When a hash output is not in agreement with the value of the digest field 187, completeness checking serves as a rejection and tunnel connection is closed. The field 188 contains the enciphered network layer datagram, for example, the enciphered IP datagram.

[0051]The encapsulated datagram is enciphered using various encoding technology. According to the embodiment of the system of this invention, the datagram 190 is enciphered to all the data also conveniently transmitted into the life of tunnel connection using a stream cipher or code block chain encryption. This becomes possible by the reliability of the transport layer connection in tunnel connection. A specific form of the symmetrical cryptographic key peculiar to encryption and connection used is determined using the stage shown in drawing 5. The field of tunnel data frames other than datagram 188 encapsulated is called a tunnel data frame header field.

[0052]Drawing 10 is a block diagram showing the embodiment of a "closing connection" frame. The field 190 contains the length of a frame. The field 191 includes the frame form which has a value equal to CLOSE (closing), for example. The field 192 contains a value equal to the present protocol version number of a tunnel protocol. The field 193 contains the state code which directs why tunnel connection is closed.

[0053]When it determines that the end point of tunnel connection should close tunnel connection working [the system of this invention], the closing connection frame shown in drawing 10 is transmitted to the end point of another side of tunnel connection. If the closing frame of closing connection is received, a receiver will close tunnel connection and, as for neither transmission nor reception, the data beyond it will be carried out through tunnel connection.

[0054]Drawing 11 is a constitutional diagram showing the embodiment which forms tunnel connection in the node which starts tunnel connection. A state is shown with an ellipse form and operation and a phenomenon are shown by the rectangle in drawing 11, and 12 and 13. For example, the tunnel server node A shown in drawing 3 works as a start side of tunnel connection, when establishing the tunnel server node D and tunnel connection. Similarly, the client system 247 of drawing 14 works as an opener of tunnel connection, when establishing the tunnel connection with a tunnel server node. A tunnel opener starts in the idle state 194. Answering the input from the user who directs that tunnel connection should be established, a tunnel opener changes from the idle state 194 to the TCP open condition 195. In this TCP open condition 195, a tunnel opener establishes the positive transport layer connection with the 1st node in alignment with a tunnel course. For example, a tunnel opener opens the socket interface relevant to the TCP connection to the 1st node in alignment with a tunnel course. In drawing 3, the node A opens the socket interface relevant to TCP connection with the node B.

[0055]A tunnel opener goes into "relay transmission" state 197 following establishment of the positive transport layer connection in the TCP open condition 195. In the this "relay transmission" state 197, a tunnel opener transmits a relay frame through positive transport layer connection in 198. A tunnel opener goes into the connected state 199 following transmission of a relay frame. When a transmission error arises during transmission of a relay frame, a tunnel opener goes into the die INGU state 208 the network error condition 215 and after that. In the die INGU state 208, a tunnel opener cuts the positive transport layer connection formed by the TCP open condition 195, for example by

cutting TCP connection with the node B. A tunnel opener goes into the dead state 210 following cutting by 209. Then, when a tunnel opener is beforehand determined by system secrecy constituting parameters, he changes and returns to the idle state 194.
[0056]In the connected state 199, a tunnel opener transmits key exchange / corroboration demand frame to a tunnel server in 200. A tunnel opener goes into the prompt machine state 201 following transmission of key exchange / corroboration demand frame 200. A tunnel opener stops at the prompt machine state 201 until he receives key exchange / corroboration response frame 202 from a tunnel server. After key exchange / corroboration response frame is received in 202, a tunnel opener goes into the authorized state 203, and transmission or reception of a tunnel data frame is performed. If a closing connection frame is received by 216 in the authorized state 203, a tunnel opener will shift to the die INGU state 208.
[0057]After a session cryptographic key is completed in 211, a tunnel opener goes into the re connection state 212, transmits a closing connection frame in 213, and cuts TCP connection with the 1st node in alignment with a tunnel course in 214. After that, a tunnel opener goes into the TCP open condition 195.
[0058]If it is detected by 205 that the data frame which the local user generated the session quit command in 204 between the authorized states 203, or was received has a corroboration error or a code error, a tunnel opener will go into the close status 206. Between the close status 206, a tunnel opener transmits a closing connection frame to a tunnel server in 207. Subsequently, a tunnel opener goes into the die INGU state 208.
[0059]Drawing 12 is a constitutional diagram showing the state in the tunnel server D, for example, the node of drawing 3, or the tunnel server 253 of drawing 14. A tunnel server starts in the acceptance waiting state 217. In the acceptance waiting state 217, a tunnel server receives the demand 218 of positive transport layer connection, for example, a TCP connection demand, from the node C of the last of the tunnel course in front of a tunnel server, for example, the node of drawing 3. The TCP connection demand 218 is answered, a tunnel server accepts the demand, and the socket interface relevant to TCP connection with the node C which this produces is established.
[0060]If TCP connection with the node of the last of the tunnel course in front of a tunnel server is established, a tunnel server will go into the relay receive state 219. In the relay receive state 219, when a tunnel server stands by so that a relay frame may be received in 220, and it is received, it goes into the connection waiting state 221. When a certain network error 234 occurs during reception of the relay frame in 219, a tunnel server goes into the die INGU state 230. Between the die INGU states 230, a tunnel server cuts the transport layer connection with the node of the last of the tunnel course in front of a tunnel server in 231. A tunnel server goes into the dead state 232 after cutting of connection.
[0061]In the connection waiting state 221, a tunnel server stands by reception of key exchange / corroboration demand frame in 222. A tunnel server determines whether the demanded tunnel connection is permitted in Step 223 following reception of key exchange / corroboration demand frame in 222. The determination of whether tunnel connection is permitted uses a tunnel opener's name, and key exchange in key exchange / corroboration demand frame as a base.
[0062]When the demanded tunnel connection is permitted, a tunnel server returns a tunnel opener key exchange / corroboration response frame in 224. When the demanded tunnel connection is not permitted, a tunnel server goes into the close status 228, and transmits a closing connection frame to a tunnel client in 229. A tunnel server goes into the die INGU state 230 following transmission of the closing connection frame in 229.
[0063]When the demanded tunnel connection is judged that a permission is granted in Step 223, a tunnel server goes into the authorized state 225. In this authorized state, a tunnel server transmits and receives a tunnel data frame between itself and a tunnel opener. In the authorized state 225, if a tunnel server receives a closing connection frame in 233, a tunnel server will shift to the die INGU state 230. In the authorized state 225, if a session quit command is received from a user in 226, a tunnel server will shift to the close status 228, and will transmit a closing connection frame to a tunnel opener in 229. In the authorized state 225, a tunnel server will shift to the close status 228, if the completeness defect of a receive packet is detected. In the close status 228, a tunnel server transmits the closing connection frame 229, and goes into the die INGU state 230 after that.

[0064]Drawing 13 is a constitutional diagram which illustrates the state machine in a tunnel relay node. A tunnel relay node starts in the acceptance waiting state 235. If the demand which forms positive transport layer connection is received in 236, positive transport layer connection will be accepted by the requestor-side node. For example, TCP connection is accepted between a relay node and the node of this side in a tunnel course.
[0065]Subsequently, a relay node shifts to the relay receive state 237. Between the relay receive states 237, a relay node receives a relay frame in 238. Following reception of the relay frame in 238, a relay node determines what kind of advance address should be used in order to advance the frame received from the TCP connection which answered the TCP connection phenomenon 236 and was established. When the following node in a tunnel course is a tunnel server, An advance address is chosen in 239 so that the server which is operating may be chosen, when the low tunnel server of availability is chosen from the group of an usable tunnel server or other things are not operating.
[0066]A relay node goes into the advance connected state 240 following the determination of the advance address in Step 239. In this advance connected state 240, a relay node establishes the positive transport layer connection with the node (one or more) directed by the advance address (one or more) determined at Step 239.
[0067]A tunnel relay node goes into the forward condition 242 following establishment of new connection with the phenomenon 241. In this forward condition 242, a relay node advances all the frames between the connection established by 236, and the connection established by 241. If the frame which a network error is detected in 243 or directs stoppage of tunnel connection is received, a tunnel relay node will go into the die INGU state 244. A relay node cuts the connection established with the phenomenon 241 following the die INGU state 244. Subsequently, a relay node goes into the dead state 246.
[0068]Drawing 14 illustrates the tunnel connection between the tunnel client 247 and the tunnel server 253 covering the virtual private network 249 formed by the pseudo network adaptor 248, and the public network 251. The tunnel server 253 and the tunnel client 247 are CPU or a microprocessor, a memory, and a network station containing various I/O devices, for example. Through the physical network adaptor 254, it is physically connected to private LAN256 including the network node 1 (257) and the network node 2 (258), and the tunnel server 253 is shown. It is further connected to the firewall 252 which separates private LAN256 from the public network 251 physically, and the tunnel server 253 is shown. The firewall 252 is physically connected to the public network 251. The tunnel server 253 is further shown that the pseudo network adaptor 255 is included. It indicates that the client system 247 contains the physical network adaptor 250 physically connected to the public network 251.
[0069]To the tunnel client 247, the node in the virtual private network 249 looks working [the element shown in drawing 14] as if it was physically connected to the client system through the pseudo network adaptor 248. The transmission between a tunnel client and the node which seems to be in a virtual private network is sent through the pseudo network adaptor 248. Data transmission between the tunnel client 247 and the tunnel server 253 is physically performed using the tunnel connection between the tunnel client 247 and the tunnel server 253.
[0070]Drawing 15 illustrates the element of a pseudo network adaptor like the pseudo network adaptor 248 of drawing 14. In one embodiment, the element shown in drawing 15 is carried out as software performed in the tunnel client 247 shown in drawing 14. As shown in drawing 15, the pseudo network adaptor 259 is provided with the following.
Virtual adapter driver interface 263.
Encapsulation engine 264.
Encryption engine 265.
The capsule release engine 268 and the decryption engine 266.
The ARP server emulator 270 and the dynamic host configuration protocol (DHCP) server emulator 272 are also shown in the pseudo network adaptor 259.
[0071]Through the virtual adapter driver interface 263, the pseudo network adaptor 259 interfaces with the TCP/IP protocol stack 260, and is shown. The TCP/IP protocol stack 260 interfaces with other services of the operating system 261, and the driver 262 of a physical network adaptor, and is shown. The driver 262 of a physical network adaptor is a device driver which controls operation of a

physical network adaptor like the physical network adaptor 250 shown in drawing 14, for example.
[0072]It registers into the network layer of the TCP/IP stack 260 that the pseudo network adaptor 259 can reach the IP address of the node in the virtual private network 249 shown in drawing 14 working [the element shown in drawing 15]. For example, it registers that the pseudo network adaptor of a client system can reach the pseudo network adaptor of a server. Then, a TCP/IP stack lets the message from the tunnel client by which the address was carried out to the node which can reach through a virtual private network pass to the pseudo network adaptor 259. The pseudo network adaptor 259 ranks second, and enciphers a message, and encapsulates a message to a tunnel data frame. The pseudo network adaptor 259 ranks second, returns a tunnel data frame to the TCP/IP stack 260, and transmits to the physical network adaptor of a tunnel server. Through and this carry out capsule release of the message to the pseudo network adaptor of a server, and a tunnel server deciphers the received data frame to it.
[0073]Drawing 16 illustrates the pseudo network adaptor 280 in detail. The pseudo network adaptor 280 is provided with the virtual network adaptor driver interface 288. The transmission path 290 is provided with the encryption engine 292 and the encapsulation engine 294. The encapsulation engine 294 interfaces with the TCP/IP transmitting interface 312 in a TCP/IP protocol stack, For example, when there is no relay node in the 1st relay node of a tunnel course, or a tunnel course, it interfaces with the socket interface in relation to a remote tunnel end point.
[0074]In the embodiment of drawing 16, the pseudo network adaptor 280 appears like an Ethernet adaptor for the TCP/IP protocol stack 282. Therefore, Ethernet packet 286 of the destination address understood to be able to reach through a virtual private network by a TCP/IP protocol stack, Virtual network adaptor interface 288 navel is carried out from the TCP/IP protocol stack 282, and it is sent through the transmission path 290. Similarly, the virtual network adaptor interface 288 navel of Ethernet packet 284 received through the pseudo network adaptor 280 is carried out from the received path 296, and it is sent to the TCP/IP protocol stack 282.
[0075]The received path 296 shown in the pseudo network adaptor 280 of drawing 16 is provided with the following.
The decryption engine 298 which interfaced with the virtual network adaptor interface 288.
Capsule release engine 300.
The capsule release engine 300 ranks second and interfaces with the TCP/IP reception function 314 of the TCP/IP protocol stack 282, For example, when there is no relay node in the 1st relay node of a tunnel course, or a tunnel course, it interfaces with the socket interface in relation to a remote tunnel end point. The pseudo network adaptor 280 is further provided with the ARP server emulator 304 and the DHCP server emulator 306. It lets ARP and the DHCP request packet 302 pass respectively to the ARP server emulator 304 and the DHCP server emulator 306. When letting the received packet pass from the received path 296 to the TCP/IP stack 282, For example, a receiving phenomenon must be directed to the TPC/IP stack 282 through an interface like the network device interface specification (NDIS) defined by Microsoft Corp.
[0076]As shown in drawing 16, the operating system 310 is connected to the TCP/IP protocol stack 282. Generally the TCP/IP protocol stack 282 is considered to be a component of an operating system. Therefore, the operating systems 310 of drawing 16 are the remaining operating system functions and procedures other than TCP/IP protocol stack 282. The physical network adaptor 308 is shown that it operates by the TCP/IP protocol stack 282.
[0077]A user directs the IP address of the node by which the data for transmitting to the TCP/IP protocol stack 282 should be transmitted to through and a message to a TCP layer, for example through a socket interface working [the element shown in drawing 16]. The TCP/IP protocol stack 282 judges whether it ranks second and a destination node can be reached through a virtual private network. When a message is a thing for the node which can reach through a virtual private network, the TCP/IP protocol stack 282 sends Ethernet packet 286 corresponding to the message to the pseudo network adaptor 280. Subsequently, as for the pseudo network adaptor 280, through and an Ethernet packet are enciphered and encapsulated by the transmission path in Ethernet packet 286 to a tunnel data frame. A tunnel data frame is returned to the TCP/IP protocol stack 282 through the TCP/IP transmitting function 312, and is transmitted to a tunnel server through tunnel connection. According to the embodiment shown here, before being enciphered within the transmission path of a

pseudo network adaptor, a digest value is calculated to a tunnel data frame.
[0078]When the TCP/IP protocol stack 282 receives a packet working [the element shown in drawing 16], the remote end point, for example, the tunnel server, of TCP/IP tunnel connection, The packet answers a TCP receiving phenomenon and it lets it pass to the pseudo network adaptor 280. Subsequently, the pseudo network adaptor 280 carries out capsule release of the packet by removing a tunnel header. A pseudo network adaptor deciphers the data by which capsule release was carried out, and returns it to the TCP/IP protocol stack 282. The data sent from the pseudo network adaptor 280, The data which appeared like the Ethernet packet received from the actual physical device, and was contained in it for the TCP/IP protocol stack 282, Based on the information on the header of the Ethernet packet given by a pseudo network adaptor, it is sent to a suitable user by the TCP/IP protocol stack 282.
[0079]Drawing 17 is a flow chart which shows the stage performed by a pseudo network adaptor, for example during the packet transmission in the transmission path 290 of drawing 14. It is determined that a TCP/IP protocol stack may reach the destination node of the packet which should be transmitted through virtual LAN based on the destination IP address and network layer route table of a packet. In Step 320, it lets a packet pass from a TCP/IP protocol stack to a pseudo network adaptor. As a result, for example, in the virtual network adaptor interface 288 of drawing 16, the trigger of the transmitting routine of a false adapter is carried out.
[0080]In Step 322, the transmitting routine of a pseudo network adaptor processes the Ethernet header of the packet given by the TCP/IP stack, and removes it. In Step 324, a transmitting routine determines whether a packet is an ARP request packet. When a packet is an ARP request packet for the IP address of the node on virtual LAN like the pseudo network adaptor of a tunnel server, it continues to Step 326 after Step 324. Otherwise, it continues to Step 330 after Step 324.
[0081]In Step 326, the ARP server emulator of a pseudo network adaptor generates an ARP response packet. For example, when an ARP demand is a thing to the physical address corresponding to the IP address of the pseudo network adaptor of a tunnel server, an ARP response directs the physical address which should relate to the IP address and which has been specified [predetermined]. In Step 328, a pseudo network adaptor lets an ARP response pass to a virtual network adaptor interface. Subsequently, a virtual network adaptor interface directs the received packet to a TCP/IP protocol stack, for example using a NDIS interface. Subsequently, a TCP/IP protocol stack is processed as if it received the ARP response through the actual physical network.
[0082]In Step 330, a transmitting routine determines whether a packet is a DHCP request packet which requires the IP address of a pseudo network adaptor. If that is right, Step 332 will continue after Step 330. Otherwise, Step 334 continues after Step 330.
[0083]In Step 334, the DHCP server emulator of a pseudo network adaptor generates a DHCP response. Generally the format of DHCP is explained in DHCP REF. In Step 328, a pseudo network adaptor directs the IP address with which through and this received the DHCP response from the tunnel server to the virtual network adaptor interface, for example to the client data field of key exchange / corroboration response frame. Subsequently, a virtual network adaptor interface directs the received packet to a TCP/IP protocol stack. A TCP/IP protocol stack ranks second, and it processes a DHCP response as if it was received through the actual physical network.
[0084]A pseudo network adaptor enciphers a message using an encryption engine, and only an addressee deciphers a message and enables it to read it in Step 334. In Step 336, a pseudo network adaptor encapsulates the enciphered message to a tunnel data frame. In Step 338, a pseudo network adaptor transmits a tunnel data frame through tunnel connection using a TCP/IP protocol stack.
[0085]Drawing 18 is a flow chart which shows the stage performed by a pseudo network adaptor, for example during the packet reception in the received path 296 of drawing 14. In Step 350, a pseudo network adaptor is notified that the packet was received through tunnel connection. In Step 352, a pseudo network adaptor carries out capsule release of the received message by removing the header field of a tunnel data frame. In Step 354, a pseudo network adaptor deciphers the datagram by which capsule release was carried out from the tunnel data frame. According to the embodiment shown here, in Step 356, a pseudo network adaptor forms an Ethernet packet from the message by which capsule release was carried out. In Step 358, a pseudo network adaptor directs that the Ethernet packet was received through the virtual network adaptor interface to the TCP/IP protocol stack. This

is made to serve as if the TCP/IP protocol stack received the Ethernet packet from the actual Ethernet adaptor.

[0086]Drawing 19 shows the data flow within the transmission path in the embodiment of a pseudo network adaptor. In Step 1 (370), application sends the data which should be transmitted to the TCP protocol layer 372 in a TCP/IP protocol stack. Application lets data pass using the conventional socket interface to the TCP protocol layer 372, and it directs the destination IP address which should transmit data. Subsequently, the TCP protocol layer 372 lets data pass to the IP protocol layer 374 in a TCP/IP protocol stack. In Step 2 (376), a TCP/IP protocol stack determines which network interface should be used for reaching a destination IP address with reference to the route table 378.

[0087]In this example, since the destination IP address is a node which may reach through a virtual private network, the IP layer 374 determines that a destination IP address may be reached through a pseudo network adaptor from the route table 378. Therefore, in Step 3 (380), a TCP/IP protocol stack lets the packet containing data pass to the pseudo network adaptor 382.

[0088]In Step 4 (384), the pseudo network adaptor 382 enciphers a data packet, and encapsulates them to a tunnel data frame. Subsequently, the pseudo network adaptor 382 returns a tunnel data frame packet to the TCP protocol layer 372 in a TCP/IP protocol stack through the conventional socket interface to the tunnel connection with the 1st node of a tunnel course.

[0089]The TCP protocol layer 372 ranks second and forms the TCP layer packet which has a tunnel data frame as the data for every tunnel data frame. It lets the TCP frame pass to the IP layer 374. In Step 5 (386), it is again searched by the route table 378 and at this time. A destination IP address is an IP address relevant to the physical network adaptor of the tunnel server, therefore is judged that it may reach through the physical network adaptor 390. Therefore, in Step 6 (388), the device driver 390 of a physical network adaptor is called, and it lets a packet pass to a physical network adaptor. In Step 7 (392), a physical network adaptor transmits data to the physical network 394.

[0090]Drawing 20 is a figure showing the data flow in the embodiment of the packet reception accompanied by a pseudo network adaptor. In Step 1 (410), it arrives through the physical network 412, and is received by the physical network adaptor, and lets data pass to the physical network driver 414. The physical network driver 414 lets data pass to the pseudo network adaptor 426 through the conventional socket interface in through and Step 3 (424) in Step 2 (418) to the IP layer 420 and TCP layer 422. In Step 4 (428), the pseudo network adaptor 426, decipher and cancel [capsule] the received data, and pass TDI (transport layer DEPEN dent interface API) of a TCP/IP stack in it, for example -- it sends to IP layer of a TCP/IP protocol stack. Subsequently, a TCP/IP protocol stack lets data pass and it lets it pass to the user relevant to the destination IP address of datagram by which capsule release was carried out in Step 5 (430).

[0091]Drawing 21 shows the data flow in the embodiment of the packet transmission accompanied by a pseudo network adaptor. Drawing 21 shows the example of use in a Microsoft Windows 95(registered trademark) PC platform. In drawing 21, the user application 450 lets the deciphered data pass to the interface to the TCP layer of the TCP/IP protocol of WinSock API 452, for example. A user directs the destination IP address relevant to the node which may reach through the virtual private network which can be accessed through a pseudo network adaptor.

[0092]Through and this IP layer rank data second to the IP layer 456, and TCP layer 454 lets data pass to the network device interface specification media access control (NDIS MAC) interface 458. It has already registered that the pseudo network adaptor 459 can arrive at the gateway address relevant to the destination IP address of the user datum into the route layer (IP). Therefore, IP layer uses a NDIS MAC layer interface, and calls the virtual device driver interface 460 to the pseudo network adaptor 459. The pseudo network adaptor 459 is provided with the following.

Virtual device driver interface 460.

ARP server emulator 462.

DHCP server emulator 464.

[0093]In the embodiment of drawing 21, the pseudo network adaptor 459 lets data pass to the tunnel application program 466. The tunnel application program 466 enciphers the IP packet received from IP layer, and encapsulates it to a tunnel data frame. Subsequently, tunnel application directs the destination IP address of through and a remote tunnel end point for the tunnel data frame containing

the enciphered data to the WinSock interface 452. Subsequently, a tunnel data frame is transmitted to the network 470 through TCP layer 454, the IP layer 456, the NDIS MAC layer interface 458, and the physical layer 468. Since the packet obtained by this does not include the destination IP address registered for conveyance of a pseudo network adaptor, these packets are not supplied to a pseudo network adaptor.

[0094]Drawing 22 is a figure showing the data flow in the embodiment of the packet transmission accompanied by a pseudo network adaptor. The embodiment shown in drawing 22 is used for a UNIX platform. In drawing 22, the user application 472 directs the destination IP address of the node which may reach the deciphered data through through and a virtual private network to the socket interface to the TCP/IP protocol stack of the UNIX socket layer 474.

[0095]The UNIX socket layer 474 sends data through TCP layer 476 and the IP layer 478. It has already registered that the pseudo network adaptor 480 can reach the gateway relevant to the destination IP address of the user datum into the route layer (IP). Therefore, the IP layer 478 calls the virtual device driver interface 482 to the pseudo network adaptor 480. The IP layer 478 lets data pass to the pseudo network adaptor 480. The pseudo network adaptor 480 is provided with the virtual device driver interface 482 and the DHCP server emulator 484.

[0096]According to the embodiment of drawing 22, the pseudo network adaptor 480 sends the IP datagram which should be transmitted to UNIX Daemon486 relevant to tunnel connection. UNIX Daemon486 enciphers the IP packet received from the IP layer 478, and encapsulates them to a tunnel data frame. Subsequently, UNIXDaemon486 lets a tunnel data frame pass to the UNIX socket layer 474 through the socket relevant to tunnel connection. Subsequently, by TCP layer 476, the IP layer 478, the data link layer 488, and the physical layer 490, a tunnel data frame is processed in order to be transmitted to the network 492. Since the packet obtained by this is not addressed to the IP address registered for conveyance of the pseudo network adaptor 480, a packet is not sent to the pseudo network adaptor 480.

[0097]Drawing 23 is a flow chart which shows the stage for initializing the virtual private network by one embodiment. In the tunnel client 247 shown in drawing 14, the stage shown in drawing 23 is performed, for example. In Step 500, the tunnel application program executed by a tunnel client transmits a tunnel relay frame to a tunnel server. In Step 502, a tunnel application program transmits tunnel key exchange / corroboration demand frame to a tunnel server. The tunnel application of a tunnel server disregards the contents of the client data field in tunnel key exchange / corroboration demand frame. The tunnel application of a tunnel server fills up the client data field of tunnel key exchange / corroboration response frame with dynamic host configuration protocol (DHCP) information including the following information on a standard DHCP format, for example.

1) The IP address of the pseudo network adaptor of a tunnel client, the IP address of the pseudo network adaptor of 2 tunnel server, and the route to the node on the private network physically connected to the tunnel server which reaches through 3 tunnel connection.

[0098]In Step 504, tunnel application receives tunnel key exchange / corroboration response frame from a tunnel server. It enables it to use the client data field 508 in a tunnel connection response for the pseudo network adaptor of a tunnel client. The tunnel application of a tunnel client tells a TCP/IP stack about the pseudo network adaptor of a tunnel client being active. The pseudo network adaptor of a tunnel client is active, and ready for being initialized in Step 510.

[0099]A tunnel client system is constituted so that the IP address to the pseudo network adaptor of a tunnel client may be obtained dynamically. So, the TCP/IP stack of a tunnel client broadcasts a DHCP request packet through a pseudo network adaptor. Therefore, in Step 512, since it relates to a pseudo network adaptor, the pseudo network adaptor of a client receives the usual DHCP request packet which requires the IP address assigned dynamically from a TCP/IP stack. A pseudo network adaptor forms a DHCP response based on the client data 508 in which through and this received the DHCP request packet from tunnel application to the DHCP server emulator in a pseudo network adaptor. A DHCP response includes the IP address of the client false adapter given by the tunnel server in client data. In Step 514, a pseudo network adaptor sends a DHCP response to a TCP/IP stack.

[0100]In Step 520, tunnel application, In order to direct that all the routes to the node attached to the private network to which the tunnel server was attached can be reached only through the pseudo

network adaptor of a tunnel server, The route table in the TCP/IP stack of a tunnel client is changed. The IP address of the pseudo network adaptor of the tunnel server given to client data is specified in this way as a gateway to the node on the private network to which the tunnel server was attached. Therefore, it considers that passes through a virtual private network and these remote nodes may reach through a client pseudo network adaptor by a TCP/IP stack.

[0101]In Step 516, the pseudo network adaptor of a tunnel client receives the ARP demand for the physical address relevant to the IP address of the pseudo network adaptor of a tunnel server. A pseudo network adaptor sends an ARP demand to an ARP server emulator, and this forms the ARP response which directs the physical address of the specification which should relate to the IP address of the pseudo network adaptor of a tunnel server. In Step 518, a pseudo network adaptor lets an ARP response pass to the TCP/IP stack of a tunnel client. Answering this ARP response, a TCP/IP stack determines that the packet by which an address is carried out to one on a virtual private network of nodes must be first transmitted through a pseudo network adaptor.

[0102]In the embodiment shown here, the system of this invention specifies two physical addresses which should relate to the pseudo network adaptor of a client, and the pseudo network adaptor of a server respectively. These specified physical addresses answer the ARP demand sent through the pseudo network adaptor, and are used as a physical address respectively corresponding to the IP address for the pseudo network adaptor of a client, and the pseudo network adaptor of a server. These specified physical addresses must have the high probability which is not used for a actual network interface.

[0103]As mentioned above, although this invention was explained in detail with reference to the specific embodiment, this invention is not limited to this. If it is a person skilled in the art, change of the versatility of the above-mentioned embodiment and another embodiment of this invention will be clear from the above-mentioned explanation. Although various embodiments which use a TCP/IP protocol stack especially were described, also when other communications protocols are used, this invention can be applied effectively. Although various flow charts were shown as what has a stage performed in an illustration order, they can also carry out this invention using the stage of a different order. Although explained using a certain specific software and/or hardware platform, the same effect is attained even if it applies this invention on other platforms. So, all the change and embodiments which are included within the limits of this invention shall be covered in a claim.

[Translation done.]

* NOTICES *

DESCRIPTION OF DRAWINGS

[Brief Description of the Drawings]
[Drawing 1]It is a block diagram showing an open systems interconnection (OSI) reference model.
[Drawing 2]It is a block diagram showing a TCP/IP Internet protocol suite.
[Drawing 3]It is a block diagram showing the embodiment of tunnel connection covering the public network between two tunnel servers.
[Drawing 4]It is a flow chart which shows the embodiment of the stage performed in order to establish tunnel connection.
[Drawing 5]It is a flow chart which shows the embodiment of the stage performed in order to perform the session key management for tunnel connection.
[Drawing 6]It is a block diagram showing the embodiment of a relay frame.
[Drawing 7]It is a block diagram showing the embodiment of a connection-request frame.
[Drawing 8]It is a block diagram showing the embodiment of a connection response frame.
[Drawing 9]It is a block diagram showing the embodiment of a data frame.
[Drawing 10]It is a block diagram showing the embodiment of a closing connection frame.
[Drawing 11]It is a constitutional diagram showing the embodiment of the state machine which forms tunnel connection in the network node which starts tunnel connection.
[Drawing 12]It is a constitutional diagram showing the embodiment of the state machine which forms tunnel connection in a server computer.
[Drawing 13]It is a constitutional diagram showing the embodiment of the state machine which forms tunnel connection in a relay node.
[Drawing 14]It is a block diagram showing the embodiment of the tunnel connection between a client computer (tunnel client) and a server computer (tunnel server).
[Drawing 15]It is a block diagram showing the embodiment of a pseudo network adaptor.
[Drawing 16]It is a block diagram showing the embodiment of a pseudo network adaptor.
[Drawing 17]It is a flow chart which shows the stage performed by a pseudo network adaptor during packet transmission.
[Drawing 18]It is a flow chart which shows the stage performed by a pseudo network adaptor during packet reception.
[Drawing 19]It is a data flow figure showing the data flow in the pseudo network adaptor under packet transmission.
[Drawing 20]It is a data flow figure showing the data flow in the pseudo network adaptor under packet reception.
[Drawing 21]It is a figure showing a motion of the data enciphered in the embodiment of the system containing a pseudo network adaptor and the deciphered data.
[Drawing 22]It is a figure showing a motion of the data enciphered in the embodiment of the system containing a pseudo network adaptor and the deciphered data.
[Drawing 23]It is a flow chart which shows the stage which initializes the system containing a pseudo network adaptor.
[Description of Notations]
7 Data path
9 Physical layer

10 The 1st protocol stack
11 Data
12 Communication
13 Transmitting process
14 The 2nd protocol stack
15 Receiving process
16 Data link layer
17 Network layer
18 Data link header
20 Data link layer trailer
22 Transport control protocol (TCP)
26 IP protocol
28 Address Resolution Protocol (ARP)
32 A hardware link level and an access protocol
46 Tunnel server A
48 Private network N1
50 The 1st firewall
52 Public network
54 Tunnel repeating installation B
58 The 2nd firewall
60 Private network N2
62 Tunnel server D

[Translation done.]

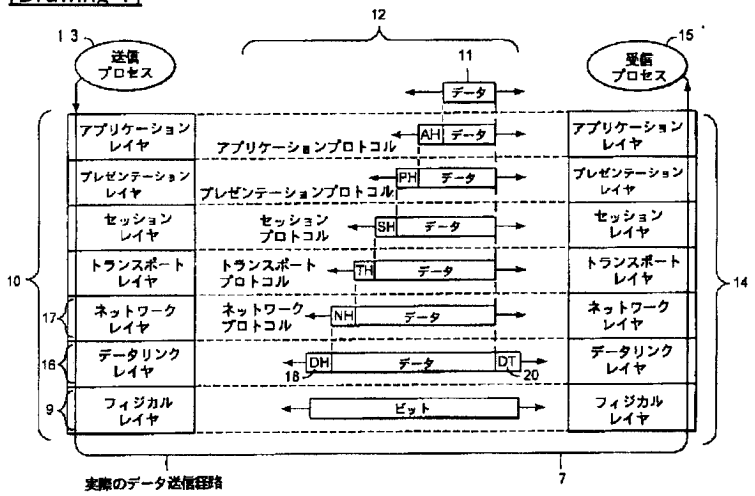
* NOTICES *

JPO and INPIT are not responsible for any damages caused by the use of this translation.

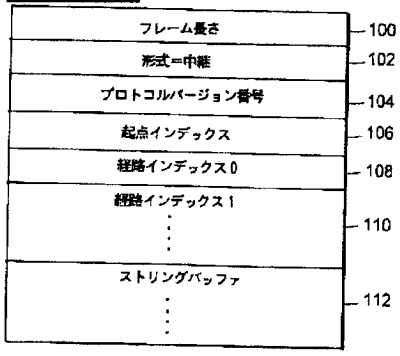
1.This document has been translated by computer. So the translation may not reflect the original precisely.
2.**** shows the word which can not be translated.
3.In the drawings, any words are not translated.

DRAWINGS

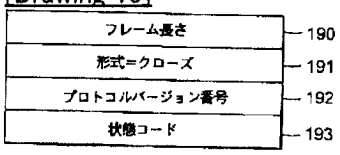
[Drawing 1]



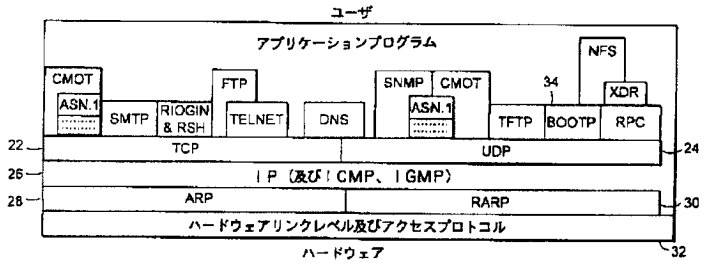
[Drawing 6]



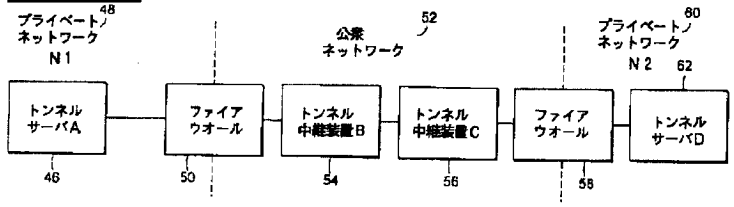
[Drawing 10]



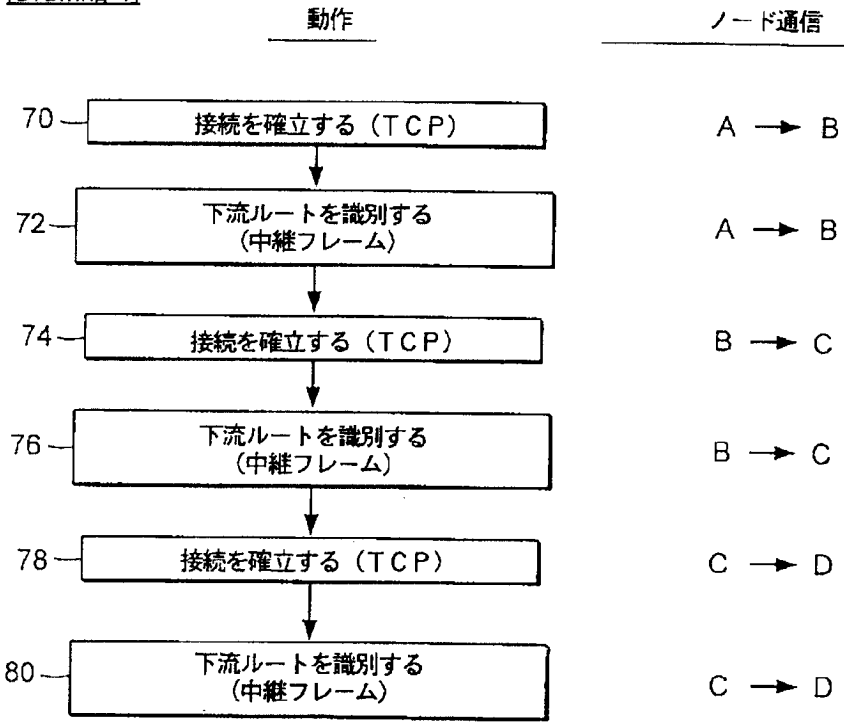
[Drawing 2]



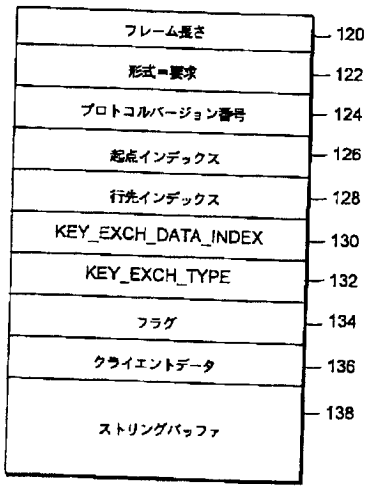
[Drawing 3]



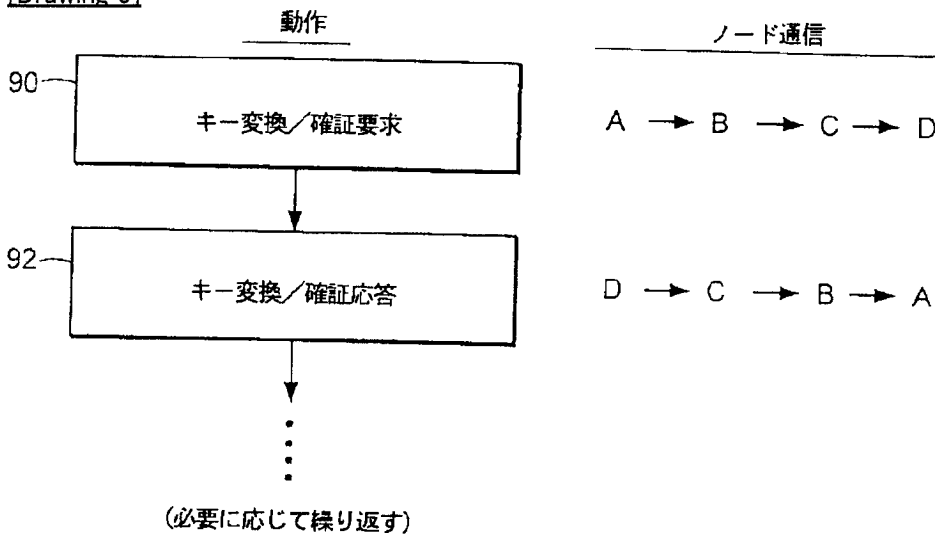
[Drawing 4]



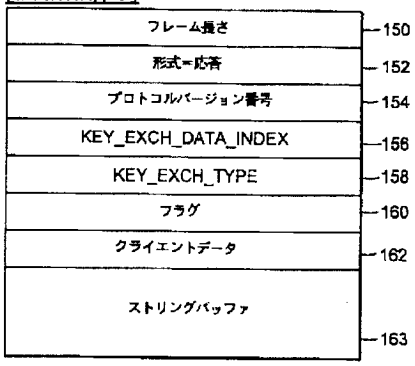
[Drawing 7]



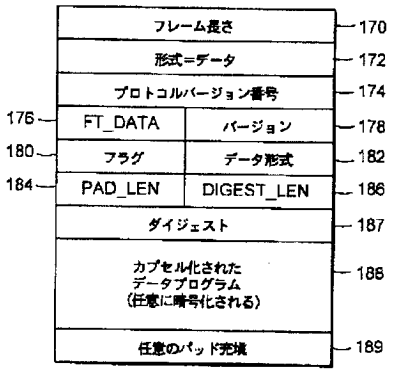
[Drawing 5]



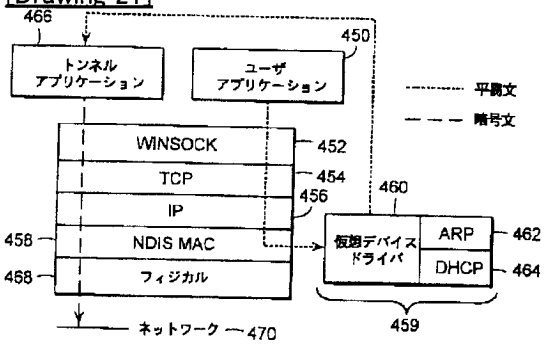
[Drawing 8]



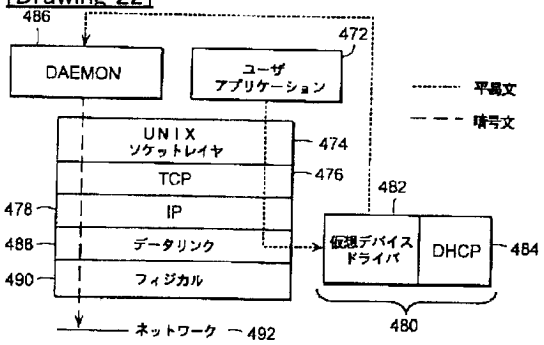
[Drawing 9]



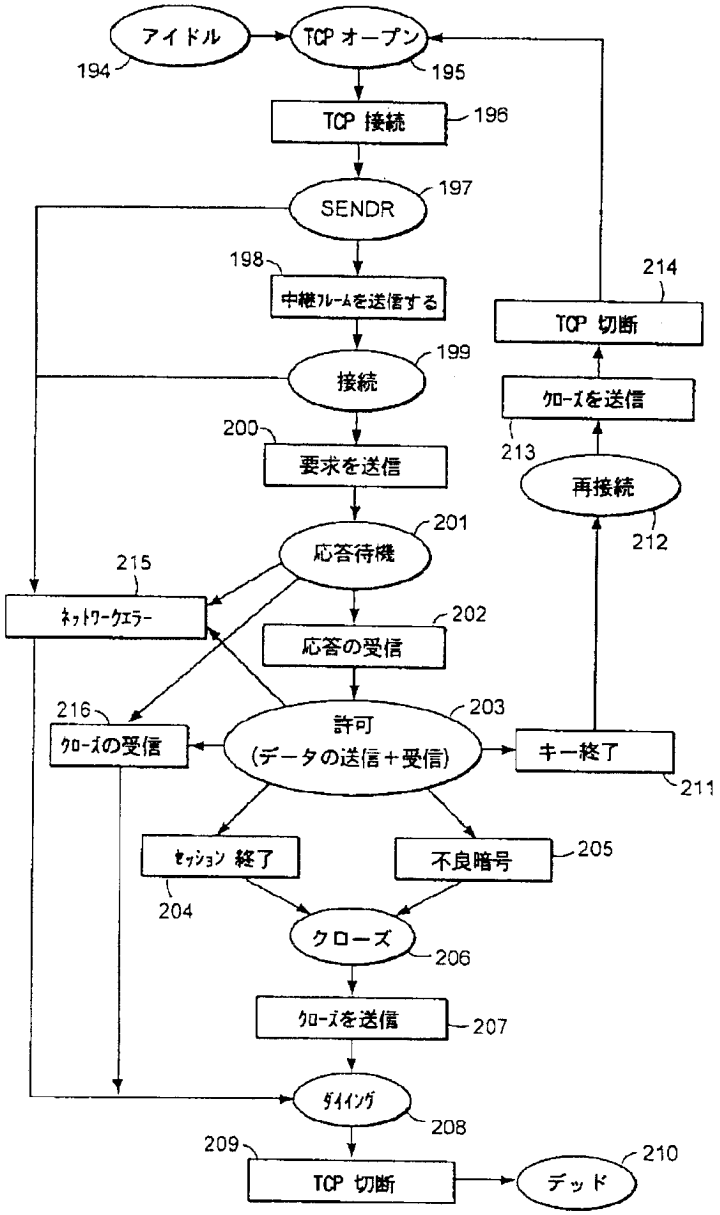
[Drawing 21]



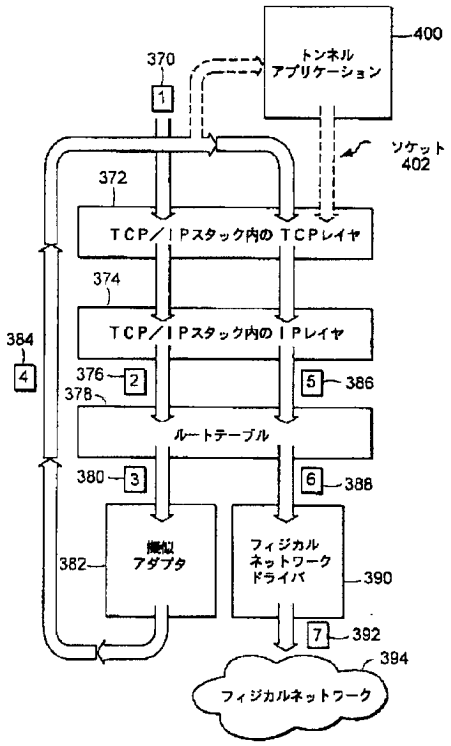
[Drawing 22]



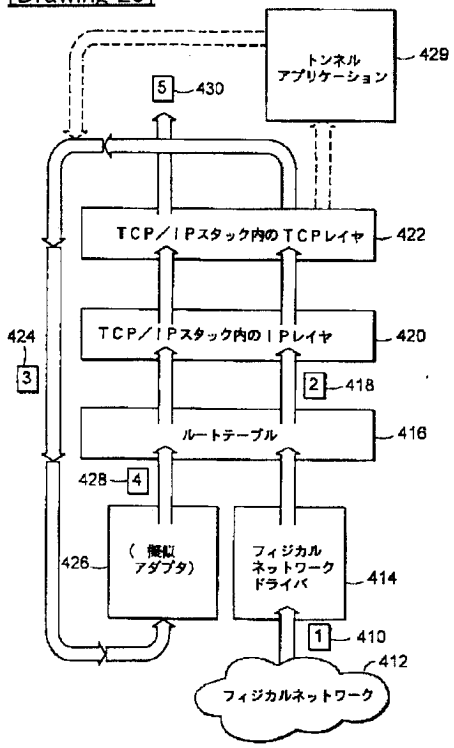
[Drawing 11]


アイドル
194
TCP オープン
195
TCP 接続
196
SENDR
197
198
中継フレームを送信する
接続
199
200
要求を送信
応答待機
201
215
ネットワークエラー
応答の受信
202
216
クローズの受信
許可
(データの送信+受信)
203
キー終了
211
214
TCP 切断
クローズを送信
213
再接続
212
セッション 終了
204
不良暗号
205
クローズ
206
クローズを送信
207
ダイイング
208
209
TCP 切断
210
デッド

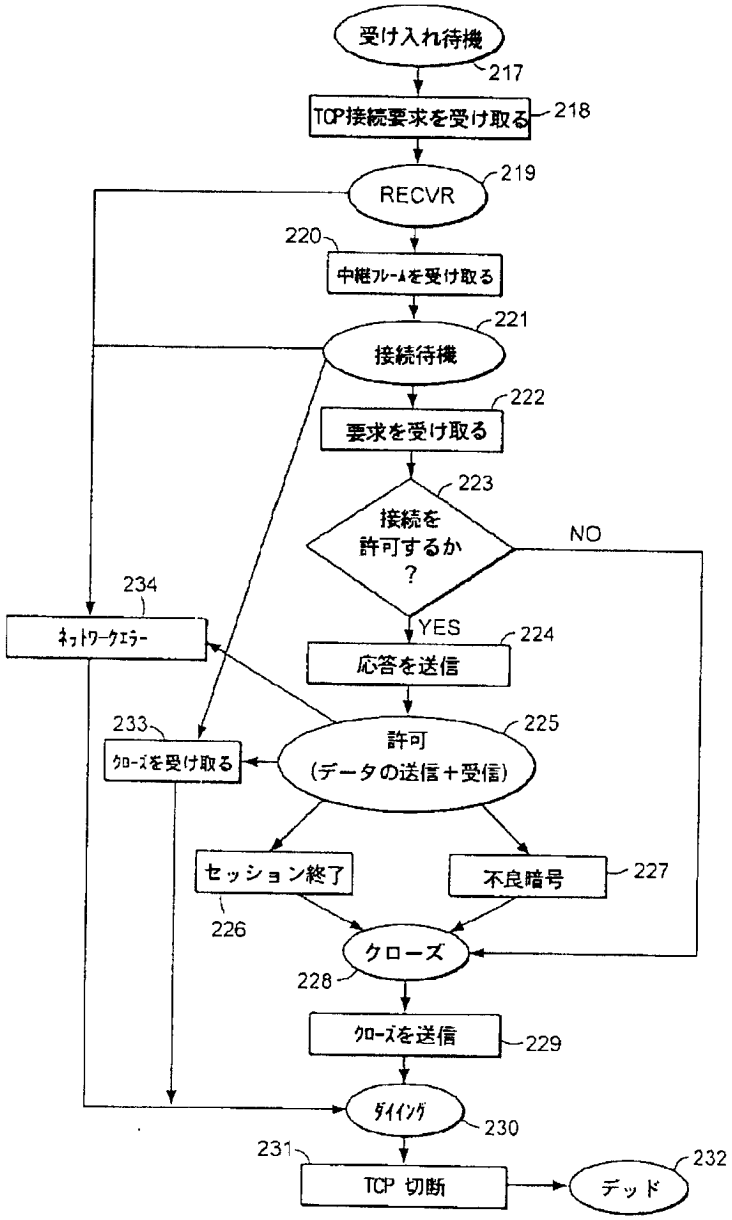
[Drawing 19]



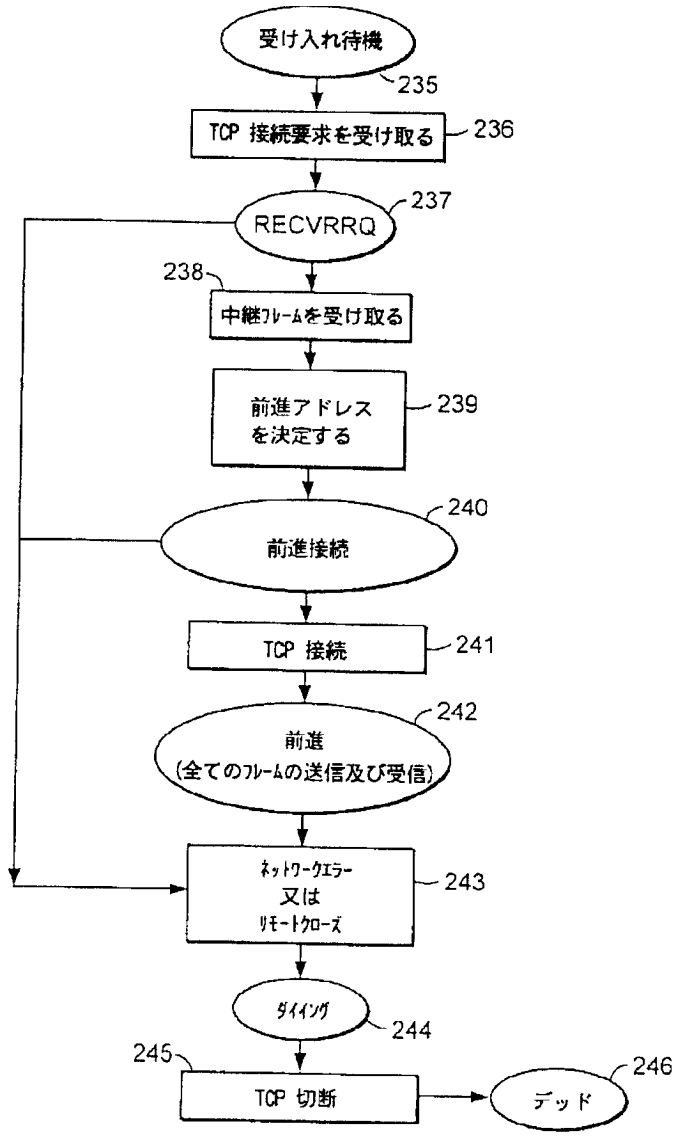
[Drawing 20]



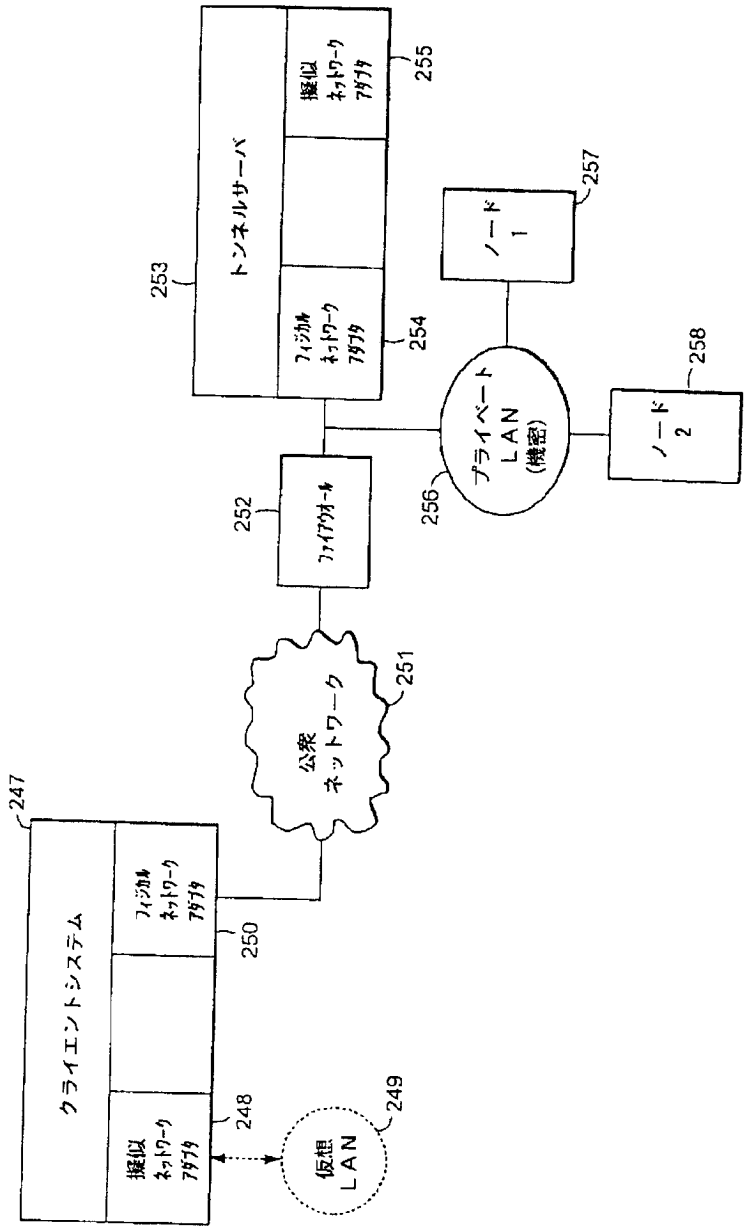
[Drawing 12]


受け入れ待機
217
TCP接続要求を受け取る
218
RECVR
219
220
中継ﾌﾚｰﾑを受け取る
接続待機
221
要求を受け取る
222
接続を
許可するか
?
223
NO
YES
234
ﾈｯﾄﾜｰｸｴﾗｰ
応答を送信
224
233
ｸﾛｰｽﾞを受け取る
許可
(データの送信＋受信)
225
セッション終了
不良暗号
227
226
クローズ
228
ｸﾛｰｽﾞを送信
229
ﾀﾞｲｲﾝｸﾞ
230
231
TCP 切断
デッド
232

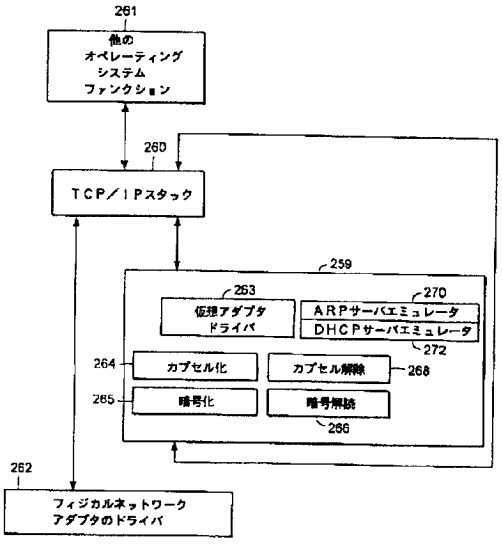
[Drawing 13]


受け入れ待機
235
TCP 接続要求を受け取る
236
RECVRRQ
237
238
中継ﾌﾚｰﾑを受け取る
前進アドレス
を決定する
239
240
前進接続
TCP 接続
241
242
前進
(全てのﾌﾚｰﾑの送信及び受信)
ﾈｯﾄﾜｰｸｴﾗｰ
又は
ﾘﾓｰﾄｸﾛｰｽﾞ
243
ﾀｲｲﾝｸﾞ
244
245
TCP 切断
246
デッド

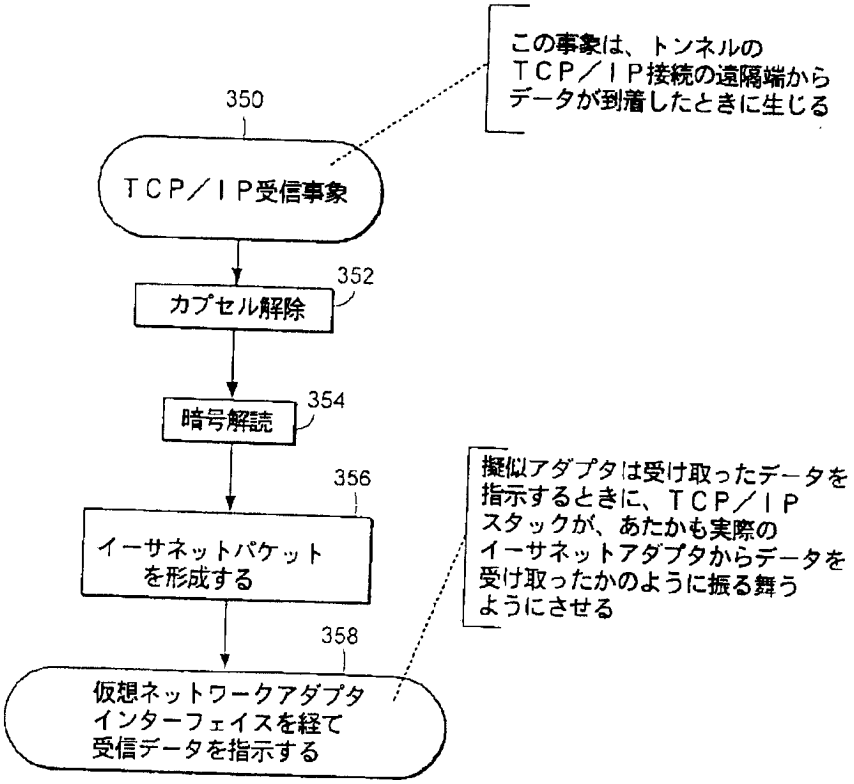
[Drawing 14]


トンネルサーバ
擬似ネットワークアダプタ
フィジカルネットワークアダプタ
253
255
254
ノード1
257
ノード2
258
プライベートLAN(機密)
256
ファイアウォール
252
公衆ネットワーク
251
クライエントシステム
247
フィジカルネットワークアダプタ
250
擬似ネットワークアダプタ
248
仮想LAN
249

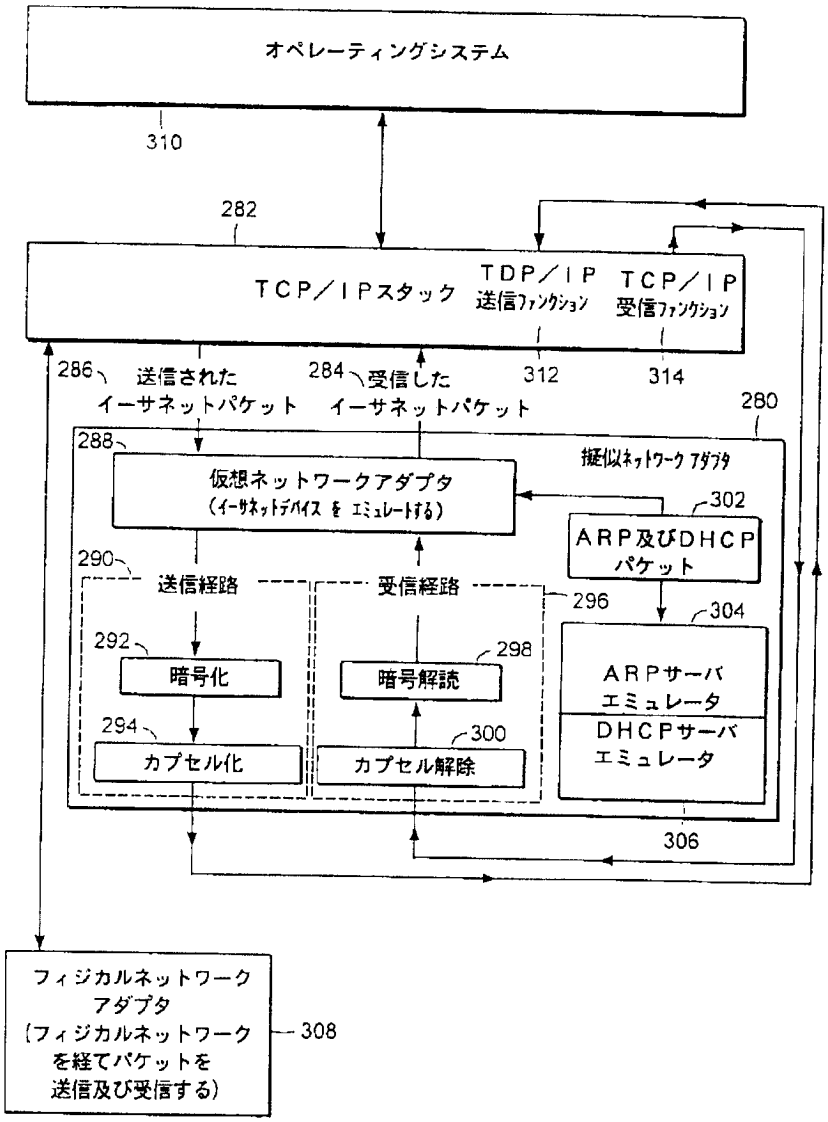
[Drawing 15]


261
他のオペレーティングシステムファンクション
260
TCP/IPスタック
259
263
仮想アダプタドライバ
270
ARPサーバエミュレータ
DHCPサーバエミュレータ
272
264
カプセル化
カプセル解除
268
265
暗号化
暗号解読
266
262
フィジカルネットワークアダプタのドライバ

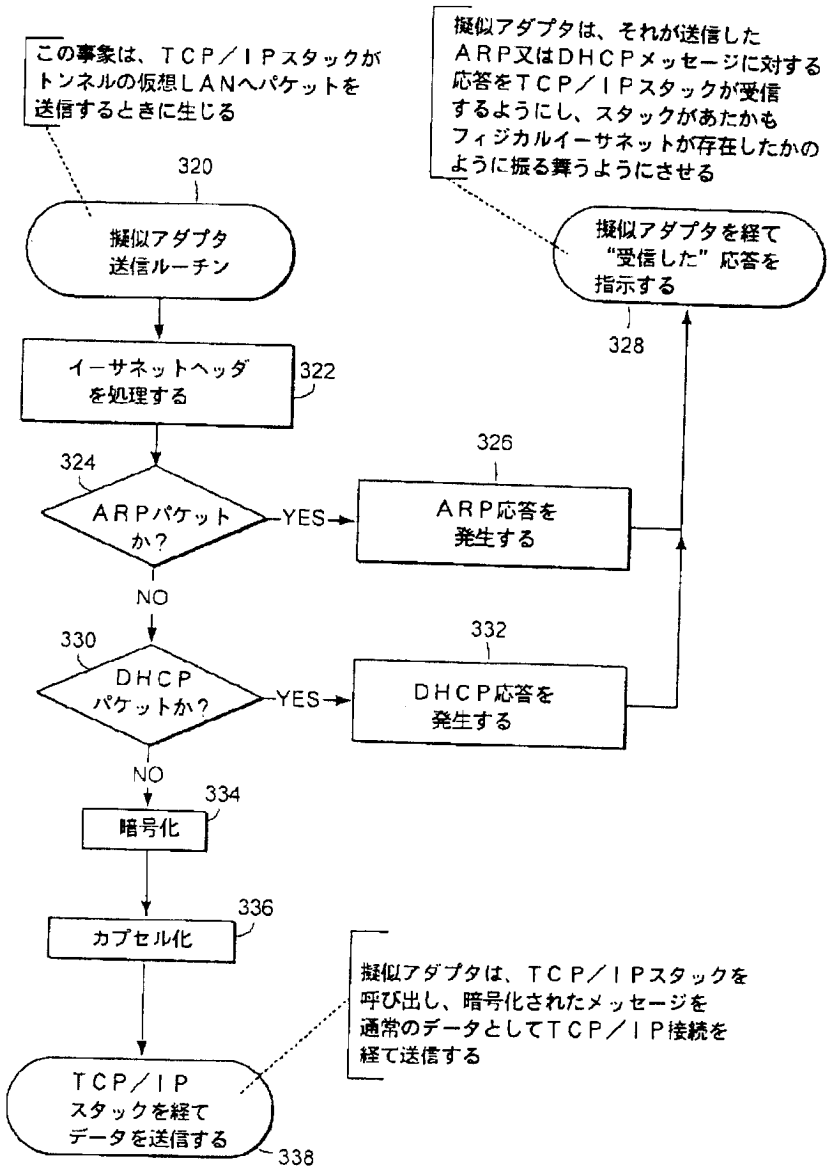
[Drawing 18]



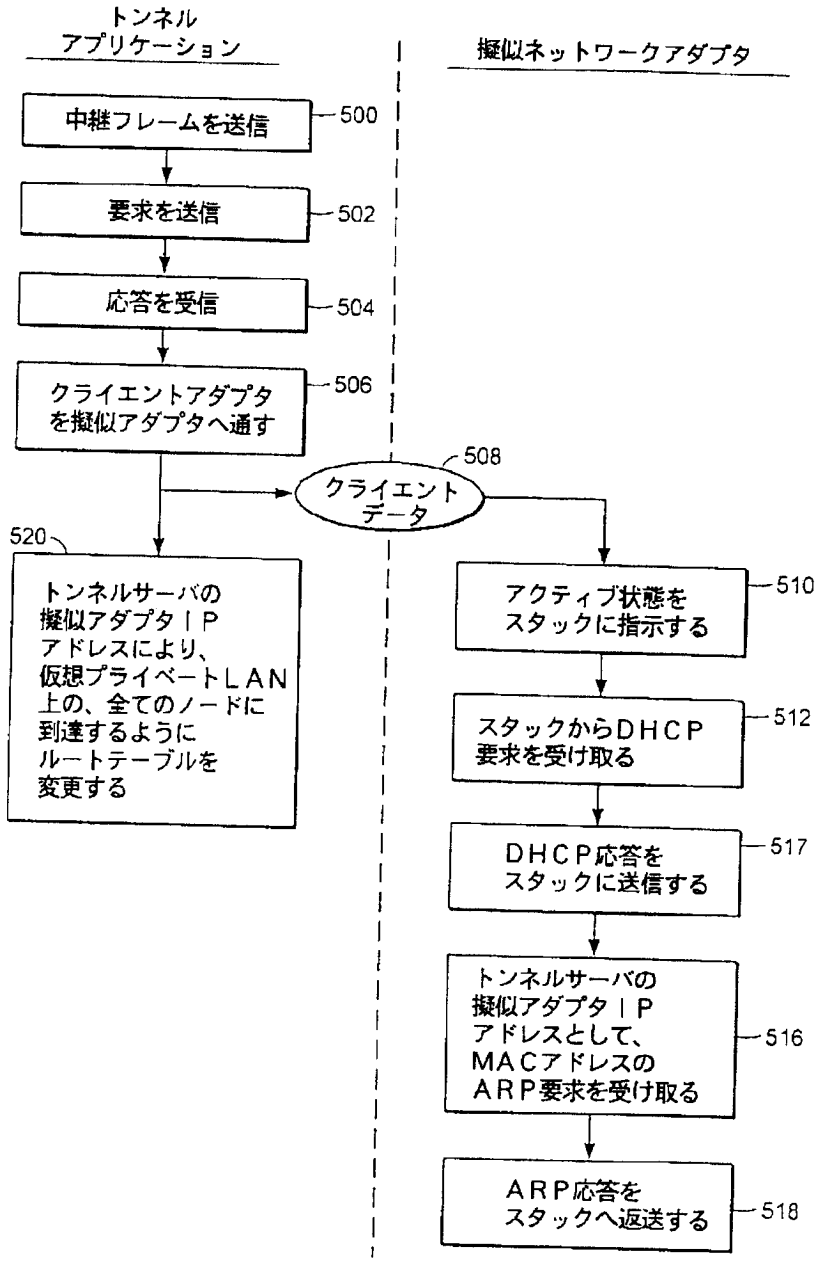
[Drawing 16]


オペレーティングシステム
310
282
TCP／IPスタック
TDP／IP 送信ファンクション
TCP／IP 受信ファンクション
312
314
286
送信された イーサネットパケット
284
受信した イーサネットパケット
280
288
擬似ネットワーク アダプタ
仮想ネットワークアダプタ (イーサネットデバイス を エミュレートする)
302
ARP及びDHCP パケット
290
送信経路
受信経路
296
304
292
暗号化
298
暗号解読
ARPサーバ エミュレータ
294
カプセル化
300
カプセル解除
DHCPサーバ エミュレータ
306
フィジカルネットワーク アダプタ (フィジカルネットワーク を経てパケットを 送信及び受信する)
308

[Drawing 17]


この事象は、ＴＣＰ／ＩＰスタックがトンネルの仮想ＬＡＮへパケットを送信するときに生じる
320
擬似アダプタ送信ルーチン
イーサネットヘッダを処理する
322
324
ＡＲＰパケットか？
YES
326
ＡＲＰ応答を発生する
NO
330
ＤＨＣＰパケットか？
YES
332
ＤＨＣＰ応答を発生する
NO
334
暗号化
336
カプセル化
ＴＣＰ／ＩＰスタックを経てデータを送信する
338
擬似アダプタは、ＴＣＰ／ＩＰスタックを呼び出し、暗号化されたメッセージを通常のデータとしてＴＣＰ／ＩＰ接続を経て送信する
擬似アダプタは、それが送信したＡＲＰ又はＤＨＣＰメッセージに対する応答をＴＣＰ／ＩＰスタックが受信するようにし、スタックがあたかもフィジカルイーサネットが存在したかのように振る舞うようにさせる
擬似アダプタを経て“受信した”応答を指示する
328

[Drawing 23]



[Translation done.]